

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Grlica

Priporočilni modul za knjižnico

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2015

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Grlica

Priporočilni modul za knjižnico

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aleksander Sadikov

Ljubljana, 2015

To delo je pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela. V primeru da se spremembe, preoblikovanja ali uporabe tega dela uporabijo v svojem delu, se lahko distribuirajo predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuirajo in/ali predelujejo pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomskega dela izdelajte priporočilni modul za lokalno knjižnico. Modul naj bo narejen kot dodatek obstoječemu knjižničnemu informacijskemu sistemu. V delu opišite splošno delovanje knjižnice za katero boste izdelali priporočilni modul. Predstavite tehnično arhitekturo aplikacije kot tudi tehnična orodja, ki bodo uporabljena za razvoj priporočilnega modula. Opišite delovanje izdelanega priporočilnega modula v realnem okolju knjižnice.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Andrej Grlica, z vpisno številko **63040410**, sem avtor diplomskega dela z naslovom:

Priporočilni modul za knjižnico

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Aleksander Sadikov,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 22. januar 2015

Podpis avtorja:

Zahvala

Za mentorstvo, posvečen čas in nasvete pri izdelavi diplomskega dela se zahvaljujem svojemu mentorju viš. pred. dr. Aleksandru Sadikov.

Zahvaljujem se staršem za podporo pri dokončanju študija.

Zahvaljujem se tudi podjetju ISA.IT d.o.o. za pridobljeno znanje v času študija.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Knjižnica	5
2.1	Vsebina knjižnice.....	5
2.2	Izposoja in vračilo knjige, periodike ali medija.....	5
2.3	Kooperativni spletni bibliografski sistem in servis (COBISS)	6
2.4	Opis delovanja aplikacije.....	9
2.4.1	Iskanje.....	9
2.4.2	Prijava.....	10
2.4.3	Zgodovina izposoje	11
2.4.4	Skrbništvo nad aplikacijo	12
2.4.5	Arhitektura aplikacije	12
2.5	Priporočila.....	13
Poglavje 3	Tehnična orodja.....	15
3.1	Tehnične specifikacije odločitvenega modula	15
3.1.1	Oracle podatkovna baza.....	15
3.1.2	Oracle Application Express	16
3.1.3	PL/SQL programski jezik.....	16
3.2	Podatkovni model	17
3.2.1	Podatkovne tabele.....	17
3.2.2	Pogledi	20
3.2.3	Paket	20
Poglavje 4	Opis delovanja odločitvenega modula	22

4.1	Nastavljeni osnovni statični parametri	22
4.1.1	Število vrstic za prikaz uporabniku	22
4.1.2	Razmerje zapisa kartona in zapisa iskanja	23
4.1.3	Razmerje zapisa podrobnega in navadnega iskanja	23
4.1.4	Razmerje zapisa priporočila in podrobnega iskanja	24
4.1.5	Utež meje podobnosti uporabnikov	24
4.1.6	Potrebno število zapisov iskanja	24
4.1.7	Potrebno število zapisov kartona izposoje	25
4.1.8	Privzet osnovni prikaz priporočila	25
4.2	Diagram poteka priporočila	26
4.2.1	Določanje števila in tipa prikaza vsebine	27
4.2.2	Izvedba odločitve priporočila	27
4.2.3	Prikaz odločitve	28
Poglavje 5	Podroben opis funkcionalnosti priporočila	31
5.1.1	Razlike med zapisi	31
5.1.2	Razlike med uporabniki	34
5.1.3	Iskanje števil prikaza različnih tipov vsebin	35
5.1.4	Iskanje podobnih uporabnikov	37
5.1.5	Iskanje najbolj ocenjenih vsebin	39
Poglavje 6	Sklepne ugotovitve	43

Seznam uporabljenih kratic

kratica	Angleško	Slovensko
APEX	Oracle Application Express	Orodje za razvoj spletnih aplikacij
CD	Compact disc	Zgoščenska
COBIB	Combinatorial Optimization: Annotated Bibliographies	Vzajemni katalog slovenskih knjižnic v sistemu COBISS
COBISS	Cooperative Online Bibliographic Systems and Services	Kooperativni spletni bibliografski sistem in servis
COLIB	Cooperative Online Library	Iskanje podatkov o slovenskih knjižnicah, vključenih v sistem COBISS
COMARC/H	Cooperative Machine-Readable Cataloging	Format za podatke o zalogi
CONOR	Cooperative Online Normative	Normativna datoteka osebnih in korporativnih imen v sistemu COBISS
DBMS_SQL	Database Management System	Podatkovni paket za upravljanje s podatkovno bazo
DDL	Data Definition Language	Podatkovno usmerjen jezik
DVD	Digital Video Disc	Digitalni pomnilniški medij
ISBN	International Standard Books Number	Mednarodna standardna knjižna številka
ISSN	International Standard Serial Number	Registracija števila serijskih publikacij
IZUM	Institute of information science, Maribor	Institut informacijskih znanosti, Maribor
OPAC	Online Public Access Catalog	Javno dostopen spletni katalog

PL/SQL	Procedural Language / Structured Query Language	Programski jezik za razvoj aplikacij na Oracle podatkovni bazi
RCUM	The Computer Centre of the University of Maribor	Računalniški center Univerze v Mariboru
SICRIS	Slovenian Current Research Information System	informacijski sistem o raziskovalni dejavnosti v Sloveniji
SZTI	The system of scientific and technical information	Sistem znanstvenega in tehničnega informiranja
SQL	Structured Query Language	Strukturirani povpraševalni jezik
YUBIB	Yugoslav Bibliographic Database	Stari knjižnični informacijski sistem, predhodnik COBISS-a

Povzetek

V diplomskem delu je predstavljen priporočilni modul, ki se vgradi v že obstoječo knjižnico. Ideja o potrebi priporočilnega modula se je rodila v Slovenskem kulturnem društvu »Bazovica« z željo, da z uvedbo priporočil izboljšajo število izposoj in število obiskov članov znotraj knjižnice.

V knjižnici si lahko člani poleg knjig izposodijo periodične tiske in različne medije, kot so npr. CD, DVD. Odločitveni modul članu priporoča knjigo, periodiko ali medij glede na njegovo zgodovino izposojanja in brskanja. Za priporočilo priporočilni modul uporablja metode primerjanja s podobnim članom in iskanje največkrat izposojenega medija v sorazmerju z oceno izdelka.

Končna priporočila, ki jih izdela priporočilni modul, so prikazana na spletni strani znotraj aplikacije »knjižnica«. Priporočilo se izdela za vsakega obiskovalca posebej in prikaže na zasebni spletni strani.

Ker knjižnica deluje na podatkovni bazi Oracle v spletnem orodju APEX, smo za dosego cilja tudi sami uporabili enako tehnologijo in razvojno orodje.

Ključne besede:

knjižnica, priporočilo, odločitev, spletna aplikacija

Abstract

Thesis describes a decision-making module which is embedded into an already existing online library. The idea was born in the Cultural and Educational Society “Slovenski dom Bazovica” in Rijeka, Croatia with a wish to introduce the recommendations in order to improve the number of borrowings and the number of members' visits within the library.

Besides books, the members can also borrow periodicals and other media such as CDs and DVDs. The decision-making module recommends books, periodicals and some other media according to member's history of borrowings and browsing. In order to give the best recommendation, the module uses methods of comparison with similar members and methods of searching the most borrowed media in proportion with the evaluation of the product.

The final recommendation done by the decision-making module is shown on the web page in the application library. The recommendations are done for each member individually and are shown on the separate web page.

Due to the fact that the library works on a database Oracle in the web development tool APEX, we have also used the same technology and the same web development tool.

Key words:

library, recommendations, decision-making, web application

Poglavje 1 Uvod

V diplomskem delu je na kratko opisano delovanje in implementacija priporočilnega modula na že obstoječo spletno knjižnico (knjižnica že deluje znotraj Slovenskega Kulturnega društva »Bazovica«). Delo na društvu je prostovoljno in zato oseba, ki je zadolžena za knjižnico, ni vedno na voljo. Član društva tako težko izve podrobne informacije o posamezni vsebini. Trenutna aplikacija deluje znotraj društva, dostop do nje imajo samo člani znotraj društva. Zaradi tega je prišlo do večjega števila pritožb na način iskanja vsebine. Kljub temu, da ima član vpogled v vso vsebino v knjižnici, o njej ne izve ničesar.

Ideja o potrebi odločitvenega modula se je rodila v samem društvu z željo po izboljšanju števila obiskov članov, števila izposoje, vsebine knjižnice in povečani možnosti pridobitve kakšnega novega člana v društvo. Z uvedbo priporočilnega modula in dovoljenim dostopom do aplikacije čez splet smo uporabnikom omogočili boljši vpogled v vsebino knjižnice, skrajšali čas iskanja vsebine in prikazali priporočila, ki so najboljše za vsakega uporabnika posebej.

Za boljše razumevanje delovanja odločitvenega modula se bomo najprej lotili opisa trenutnega stanja knjižnice in aplikacije, ki je zadolžena za arhiv vsebine knjižnice. V poglavju Knjižnica bomo najprej pogledali, kakšna sploh je vsebina knjižnice, kako je razdeljena in, kaj si lahko uporabnik izposodi. Obdelali bomo tudi problematiko, čemu služi aplikacija »knjižnica«, če že obstaja boljši in zmogljivejši sistem, imenovan COBISS. Pregledali bomo načine, kako lahko brskamo po spletni knjižnici, kako se prijavimo v aplikacijo, izposodimo vsebino ali iščemo po spletni strani. Navedli bomo tudi glavno razliko med navadnim uporabnikom in skrbnikom. V arhitekturi aplikacije bo opisan način preverjanja uporabnika, kratki opisi tabel, ki se nahajajo znotraj same podatkovne baze in vpogledi, ki pospešijo iskanje določenih zahtev uporabnikov.

V poglavju Tehnične specifikacije odločitvenega modula bomo razložili, na kakšnem sistemu deluje aplikacija ter kakšno orodje in jezike smo uporabili za izdelavo odločitvenega modula. Nekaj več besed bo posvečeno Oracle podatkovni bazi, saj na njej sloni celotna aplikacija. Trenutna aplikacija »knjižnica« deluje znotraj okolja Application Express, ki ga skrajšano imenujemo APEX. V povezovalne namene APEX-a in Oracle podatkovne baze uporabljamo

programski jezik PL/SQL. V nadaljevanju poglavja bomo prikazali celoten podatkovni model, ki je potreben za delovanje odločitvene skripte.

Po predstavitvi modela odločitvenega modula sledi poglavje Opis delovanja odločitvenega modula, v katerem bomo okvirno pregledali delovanje priporočilne skripte. Ena od pomembnih vlog v delovanju skripte so nastavitve. Zato si bomo podrobneje ogledali vrednosti osnovnih nastavitvev v odločitvenem modulu. Za boljše razumevanje delovanja v tem poglavju je podan slikovni diagram, ki nam na hitro opiše postopek delovanja skripte. Zaradi lažje razlage poteka delovanja smo skripto razdelili na tri glavne procese. Prvi proces se imenuje »določanje števila in tipa prikaza vsebine«. V tem procesu bomo najprej pregledali razlike med vsebinami in kakšno vlogo imajo med seboj. Takšen proces je zelo pomemben, saj nam določi, koliko katere vsebine prikažemo uporabniku.

Drugi glavni proces v delovanju skripte je izvedba odločitve. Ko iz prejšnjega procesa izvemo, katero vsebino prikazati ter njeno količino, nam skripta sproži iskanje najboljše vsebine, ki jo lahko priporočimo uporabniku. Ker različne uporabnike zanima različna tematika izdelkov, je potrebno poiskati najbolj podobne profile uporabnikov. Iz podobnih profilov izvemo, katera vsebina je za danega uporabnika najbolj zanimiva. Lahko pride tudi do situacije, da nimamo dovolj podobnih uporabnikov oziroma da skripta vrne premajhno število vsebine, ki bi jo lahko priporočala. Tudi za takšne težave obstaja rešitev in to je klic dodatne procedure, ki opravi še dodatno iskanje najboljših ocen za vse vsebine, ki se nahajajo v podatkovni bazi.

Po zaključku procesa iskanja nadaljujemo s tretjim procesom, ki nam govori, kako priporočamo zapise na spletni strani. Zapisi, ki pridejo iz procesa izbire zapisov, se nahajajo v podatkovni tabeli, in sicer zaradi pospešitve delovanja aplikacije. V primeru, ko uporabnik ne brska po spletni strani, ni potrebe po novem računanju priporočila. S tem skrajšamo nepotrebne klice v podatkovno bazo. Ker je lahko več uporabnikov istočasno na spletni strani in sproti brska po vsebini knjižnice, bi z vsako osvežitvijo spletne strani na novo računali priporočila in preveč obremenjevali podatkovno bazo ter s tem tudi upočasnili celotno delovanje aplikacije. Prikaz zapisov v spletni aplikaciji je velikega pomena, saj če takšne zapise prikažemo uporabniku na nelep in moteč način, bo le-ta prenehal uporabljati spletno stran. Zato smo se odločili dodati posebno spletno stran za priporočila. V tem primeru za uporabnika ne bo moteče. Za tistega, ki ga priporočila zanimajo, si bo vzel čas in pogledal, kakšno je priporočilo spletne strani.

V predzadnjem poglavju Podroben opis funkcionalnosti priporočila si bomo še podrobneje ogledali posamezen proces določanja števila prikaza vsebine in različne načine izračuna odločitve. Za boljše razumevanje delovanja posameznih procesov je potrebno najprej

predelati podrobne razlike med zapisi, kot so razlike v vsebini zapisov in razlike med zgodovino zapisov. Porazdelitev vsebine je ključnega pomena, saj je potrebno izvedeti, kaj uporabnika bolj zanima, da mu takšne vsebine bolj priporočamo.

V nadaljevanju si bomo ogledali razlike med uporabniki, saj je vprašanje, za koga delamo priporočilo, ključnega pomena. Glavna razdelitev uporabnikov je na prijavljene in neprijavljene uporabnike. Za neprijavljene uporabnike nimamo zgodovine in ne vemo, kaj jih zanima, za takšne uporabnike je zelo pomemben proces, ki zgradi privzeto priporočilo. O uporabniku, ki je prijavljen v aplikacijo, vemo veliko več, saj imamo vpogled v celotno njegovo zgodovino (če jo le-ta ima).

Predelali bomo tudi proceduro iskanja posameznega števila vsebin. Ta procedura nam govori, koliko vsebine je potrebno priporočati posameznemu uporabniku. Kot smo izvedeli že iz prejšnjih podpoglavij, poleg razlike med zapisi vplivajo na samo odločanje tudi razlike med uporabniki in njihova zgodovina. Obdelali bomo tudi pravilno uporabo uteži. Vse bo prikazano na lažjem primeru, kjer bomo lahko opisali tudi razlike med zapisi. Izdelan je tudi podroben diagram poteka, po katerem bomo lažje razumeli podrobno delovanje skripte. Potrebno je tudi vedeti, kaj se zgodi, če nimamo dovolj zapisov in, kako potem pridemo do končne odločitve.

Zapisi, ki pridejo iz procedure iskanja posameznih števil vsebin, se pošljejo naprej na proceduro iskanja podobnih uporabnikov. Procedura iskanja podobnih uporabnikov ima največji pomen v odločitveni skripti. S to proceduro poiščemo podobne uporabnike z upoštevanjem uteži podobnosti. V primeru, da je več istih, zgradimo povprečne ocene izdelkov, ki so si jih izposodili drugi podobni uporabniki, in izberemo izdelke z najboljšimi ocenami. Podroben opis delovanja procedure je podan na primeru, kjer lahko natanko prikažemo potek izbire in določanja ocene. Opisana je tudi formula, ki nam zgradi povprečno oceno za posamezen izdelek.

Zadnja procedura v tem poglavju je iskanje najbolj ocenjenih vsebin. Takšna procedura ima zelo pomembno vlogo pri iskanju priporočila za neprijavljene člane. V primeru, ko procedura iskanja podobnih uporabnikov ne najde dovolj potrebnih zapisov za prikaz priporočila, se prav tako sproži procedura iskanja najbolj ocenjenih vsebin in izbere najbolj ocenjene vsebine.

Sledil bo zaključek, v katerem bomo predelali, kaj smo dodali na obstoječo aplikacijo, kako se obnaša ob delovanju in kaj lahko v prihodnosti še izboljšamo.

Poglavje 2 Knjižnica

Knjižnica je del Slovenskega kulturnega društva »Bazovica«, ki deluje od ustanovitve društva dalje in je imela vedno pomembno vlogo v tamkajšnji skupnosti. V nadaljevanju bomo najprej izvedeli, kaj vsebuje knjižnica, kdo so njeni uporabniki in kako deluje aplikacija »Knjižnica«. Aplikacija »Knjižnica« že deluje in vsebuje vse potrebno za popolno vodenje arhiva izposoje izdelkov. V diplomskem delu je bil izdelan odločitveni modul, ki se je dogradil na že obstoječo aplikacijo in tako olajšal uporabo same aplikacije.

2.1 Vsebina knjižnice

Knjižnica vsebuje največ leposlovja slovenskih in tujih avtorjev ter manjše število strokovne literature. Poleg knjig sodi v knjižnico tudi periodika, za katero ima društvo mesečno oz. dnevno naročnino. Ker se v društvu odvija mnogo dejavnosti, kot so izložbe, sestanki in podobno, smo v knjižnico dodali tudi arhiv medijev (posnetki, slike) z možnostjo izposoje.

2.2 Izposoja in vračilo knjige, periodike ali medija

Knjigo, periodiko ali medij si lahko izposodi vsak član društva, ki ima poravnano članarino za tekoče leto.

Potek izposoje:

1. Član lahko poišče knjigo, periodiko ali medij na dva načina :
 - 1.1 Na društvu v aplikaciji »Knjižnica« sam poišče knjigo, periodiko ali medij
 - 1.2 S telefonskim klicem na društvo v času delovnih ur
2. Član se sestane z zadolženo osebo za knjižnico in prevzame željen izbor

Potek vračila :

1. Član se sestane z zadolženo osebo za knjižnico in vrne izposojene izdelke
2. Ob vračilu lahko poda osebno oceno knjige, periodike ali medija

2.3 Kooperativni spletni bibliografski sistem in servis (COBISS)

Naša knjižnica ima tudi del, implementiran v sistem COBISS, v namen oglaševanja v širši javnosti. Ker si knjige v knjižnici lahko izposodijo le člani društva s COBISS sistemom, lahko na ta način privabimo tudi nove (bodoče) člane društva.

Vzajemni katalog je računalniško voden oziroma podprt centralni katalog, ki je skupni katalog več knjižnic.

OPAC je javno dostopen spletni katalog, ki ga je moč doseči izven knjižnice. Zanj, kot informacijski vir, ni več prostorske in časovne omejitve.

Razvoj COBISS-a:

1982 RCUM (kasneje se preimenuje v IZUM) in UKM zasnujeta in izpeljeta prvo računalniško podporo knjižničnih opravil. RCUM (IZUM) dobi funkcijo gostitelja SZTI (Sistema za zdravstveno in tehnično informiranje) najprej v Sloveniji, nato pa za celotno tedanjo Jugoslavijo. Oblikujejo tudi sistem/bazo YUBIB. Po osamosvojitvi Slovenije se sistem preoblikuje v COBISS. Kot sistem vzajemne katalogizacije povezuje veliko večino knjižnic v Sloveniji. Danes posreduje tudi dostop do različnih bibliografskih baz in baz s polnimi besedili. Vodi tudi SICRIS, informacijski sistem za podporo odločanju na področju raziskovalne dejavnosti Slovenije.

COBISS predstavlja organizacijski model povezovanja knjižnic v nacionalni knjižnični informacijski sistem z vzajemno katalogizacijo, vzajemno bibliografsko-katalogno bazo podatkov COBIB in lokalnimi bazami podatkov sodelujočih knjižnic, bazo podatkov o knjižnicah COLIB, normativno bazo podatkov CONOR ter s številnimi drugimi funkcijami.

COBISS/OPAC je spletna aplikacija (Slika 2.1), ki knjižnicam in končnim uporabnikom omogoča dostop do:

- bibliografsko-katalognih baz podatkov v sistemu COBISS (COBIB in lokalne baze podatkov knjižnic),
- drugih baz podatkov COBISS (npr. COLIB, CONOR),
- nekaterih specializiranih baz podatkov (domačih ali tujih proizvajalcev),
- baz podatkov na oddaljenih strežnikih Z39.50.

COBISS Virtualna knjižnica Slovenije **IZUM**

Pomoč v ŽIVO Pomoč Nastavitve slovenski

Baze podatkov Iskanje Moja knjižnica

Lokalna baza podatkov: Slovenski dom - Kulturno prosvetno društvo 'Bazovica', Rijeka (Štev. zapisov: 155)

Iskanje

Osnovno **Izbirno** Ukazno

Avtor IN
 Naslov IN
 Leto izida IN
 Ključne besede

Jezik: vsi jeziki
 Izbor zapisov: vse gradivo (tudi e-viri)
 Le viri z e-dostopom: ☐

POIŠČI

Namigi

Pregled pojmov
 Če nismo prepričani, kakšna je prava oblika imena, si lahko v izbirnem načinu iskanja pomagamo s pregledom pojmov v iskalnih indeksih. Vtipkamo začetek avtorjevega imena in kliknemo gumb (pregled pojmov). Dobili bomo seznam imen, s katerim izberemo ustrezno obliko imena. Ime se bo samodejno preneslo v iskalno okence. Če želimo, pa lahko ustrezno obliko imena vnesemo v iskalno okence tudi ročno.

Osebnostna imena
 Osebnostna imena v poljih za avtorstvo oz. poljih za ime kot predmetno oznako so invertirana - priimek je na prvem mestu, sledi pa eno ali več imen. Med priimkom in imenom stojita vejica in presledek. Pri fraznem načinu iskanja moramo vnesti ime v obliki Priimek, Ime.

Primer
 "Barto, vladimir"

[več o izbirnem iskanju](#)
[podrobno o izbirnem iskanju](#)

[NA VRH](#) | [Baze podatkov](#) | [Iskanje](#) | [Moja knjižnica](#) | [Izhod](#)

Mobilna različica | Lahki COBISS/OPAC | POGOJI UPORABE | Piškatki | COBISS/OPAC, V6.1 - Novosti | Predlogi? Pošljite jih na: cobissuser@izum.si | © IZUM, 1997-2014

Slika 2.1: Cobiss/OPAC

Postopki iskanja po bibliografsko-katalognih bazah podatkov v sistemu COBISS so prilagojeni različnim stopnjam znanja uporabnikov (osnovni, izbirni in ukazni način iskanja).

Uporabniki lahko kopirajo zapise iz baze podatkov COBIB za osebno uporabo, skladno s pogoji uporabe. Knjižnice pa lahko kopirajo zapise za uporabo v lokalnih knjižničnih aplikacijah, skladno z določili posebnih pogodb.

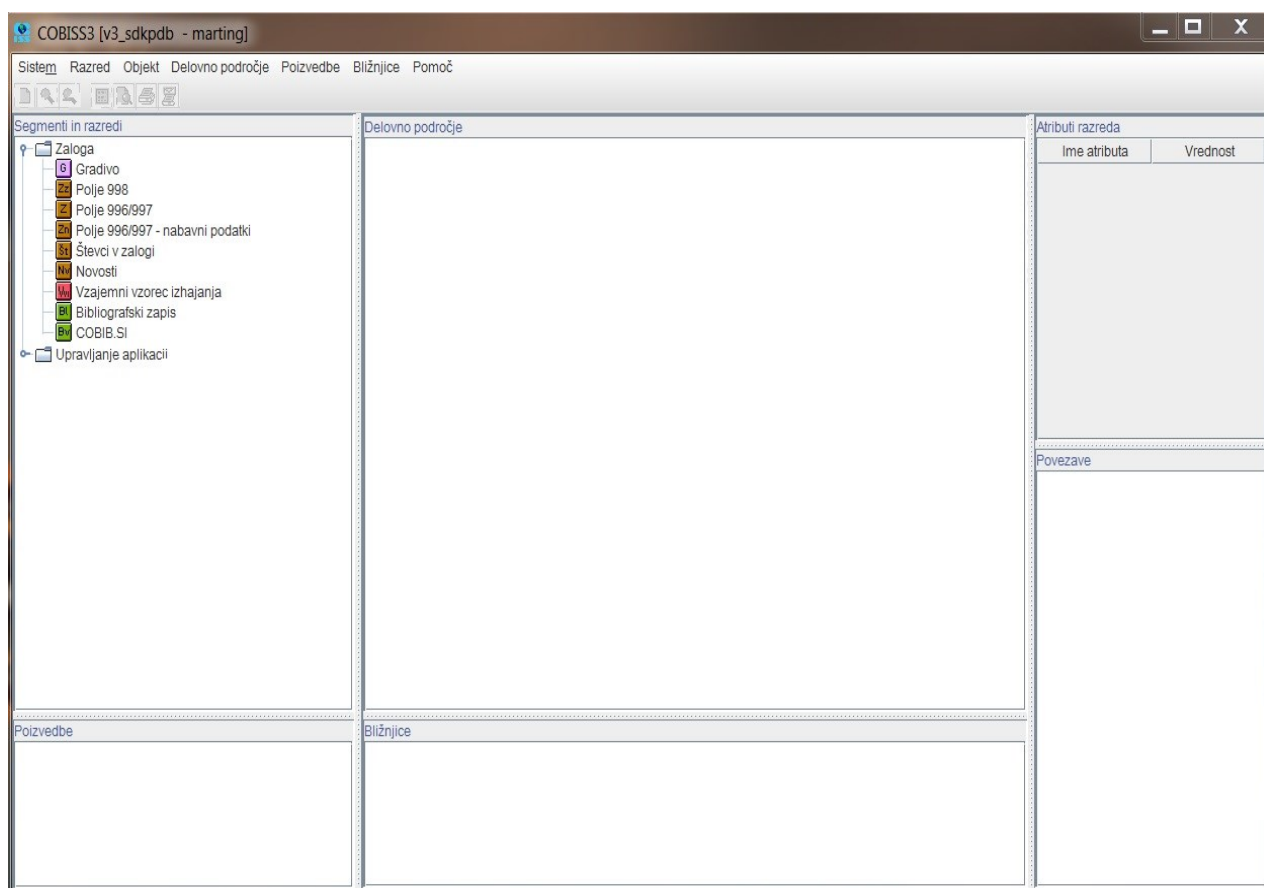
V okviru lokalnih aplikacij je veliko pozornosti posvečeno poenotenju vnosa in prikazovanja lokalnih podatkov, ki so namenjeni končnim uporabnikom. Enotnost je v precejšnji meri zagotovljena s formatom COMARC/H za podatke o zalogi. Poleg tega pa se pred vključitvijo posamezne knjižnice v sistem nacionalni center COBISS z vsako knjižnico posebej dogovori o načinu vnosa lokalnih podatkov. Način vnosa podatkov se prilagaja čim prijaznejši predstavitvi za končne uporabnike, obenem pa se upoštevajo posebnosti organizacije dela v posamezni knjižnici.

Programski segmenti v sistemu COBISS3:

- Nabava,
- Serijske publikacije,
- Zaloga,

- Izposoja,
- Medknjižnična izposoja,
- Izpisi,
- Upravljanje aplikacij.

V Slovenskem domu "Bazovica" še nimamo vključenih programskih segmentov Serijske publikacije, Izposoja in Medknjižnična izposoja. Ker smo se komaj vključili v sistem COBISS3 (Slika 2.2), uporabljamo zaenkrat samo segment Zaloga, znotraj katerega prenašamo zapise knjig iz vzajemne baze v lokalno bazo.



Slika 2.2: Cobiss3 aplikacija za Knjižnice

2.4 Opis delovanja aplikacije

V nadaljevanju bomo opisali trenutno delovanje aplikacije »Knjižnica«, ki se nahaja v kulturnem društvu »Bazovica«. Opisali bomo postopke, kot so: iskanje vsebine, prijava v aplikacijo in uporabnikova zgodovina izposoje. Nekaj več besed bo o skrbništvu in arhitekturi celotne aplikacije.

2.4.1 Iskanje

Iskanje poteka na dva različna načina – odvisno od tipa uporabnika. Dokler uporabnik ni prijavljen v aplikacijo, poteka javni pregled iskanja čez splet. Spletno poročilo je odvisno od parametrov pregleda. Podoben primer lahko pogledamo na sliki (Slika 2.3).



ID	TIP	NAZIV	ISBN	MEDIJ	AVTORJI	LETO	ZALOŽNIK	MESTO	STANJE_ZAPISA
5	Periodika	Planinski vestnik	PV-043-4343	-	-	2013	-	Ljubljana	Na zalogi
1	Knjiga	Hlapič	5453-5345-4554	-	Ivana Brič Mažuranič, Marko Skače, Miroslav Krleža, Oton Župančič	-	Školska knjiga d.d.	-	Na zalogi
2	Knjiga	Ime knjige 7	54543-534534	-	Ivana Brič Mažuranič	-	Mozak knjiga	-	Izposojeno
3	Medij	Slike Božični nastup 2013	-	CD	Ilija Dadasović, Milan Grlica	-	KPD "Bazovica"	Rijeka	Izposojeno
11	Knjiga	Nova knjiga test 66	43232-32432-4334	-	Andrej Grlica	2013	Andrijić d.o.o.	Baška	Na zalogi
12	Periodika	Hrvatski planinar	9343-4324-323-3232	-	-	-	Novi informator	Zagreb	Na zalogi
7	Periodika	Sopotje	423-234	-	-	2013	KPD "Bazovica"	Rijeka	Na zalogi
13	Periodika	Drvo znanja	1888555-333-2	-	-	-	Školska knjiga d.d.	Zagreb	Na zalogi

Slika 2.3: Primer iskanja na spletni strani

Polje »Išči« namenoma pregleda celotno poročilo in poišče podoben tekst, polje »Tip« nam omeji pregled v odvisnosti od izbire knjige, periodike ali medija. »Pregled« nam pove, koliko zapisov prikaže na trenutni spletni strani. V primeru, da kliknemo gumb »Prikaži«, se nam na spletni strani na novo naložijo in prikažejo zapisi glede na parametre iskanja. Gumb »Ponastavi« nam ponastavi vse parametre na začetno vrednost in osveži poročilo.

Ko se uporabnik prijavi v aplikacijo, ima drugačen tip pregleda, ki se imenuje *interaktivno poročilo* in nam omogoča dinamičen prikaz vsebine v odvisnosti od parametrov iskanja. Primer prikaza samo izposojenih zapisov lahko vidimo na spodnji sliki (Slika 2.4).



Slika 2.4: Primer iskanja za prijavljene uporabnike

2.4.2 Prijava

V aplikacijo se prijavimo s klikom na gumb »Uporabnik«, ki nas prestavi na spletno stran za prijavo v spletno aplikacijo knjižnica. Zgled prijavnega obrazca si lahko ogledamo na spodnji sliki (Slika 2.5).



Slika 2.5: Primer prijavnega okna v spletno aplikacijo

Pravice za prijavo v aplikacijo Knjižnica dobimo iz baze članov društva, ki vsebuje naslednje podatke: tip uporabnika, ime, priimek, datum rojstva, naslov bivanja, e-pošta, uporabniško ime, zakodirano geslo, e-naslov in ali ima poravnano članarino za tekoče leto. Pravice za prijavo v spletno aplikacijo Knjižnica lahko spreminjajo le osebe, ki imajo pravico dostopa v glavno aplikacijo članov društva.

V aplikaciji imamo dva tipa uporabnikov: navadne uporabnike in skrbnike knjižnice. Navadni uporabniki imajo le pravico pregleda vsebine v knjižnici, ogled lastne zgodovine izposojanja in prikaz lastne osnovne informacije o uporabniku, kot so: naslov prebivanja, e-naslov, plačana članarina za tekoče leto in podobno. Primer prikaza informacij o uporabniku lahko vidimo na spodnji sliki (Slika 2.6).

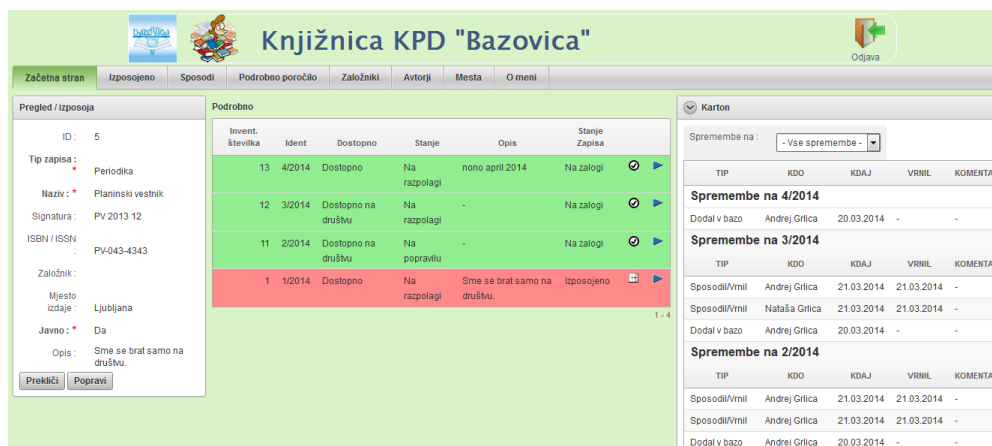


Slika 2.6: Primer prikaza informacij o spletnem uporabniku

Skrbniki knjižnice imajo na voljo vse, kar ima navaden uporabnik in še dodatno upravljanje s celotno knjižnico. Dodatno upravljanje vsebuje spreminjanje šifrantov (založniki, avtorji, mesta izdaje) kot tudi dodajanje in popravljanje vsebine knjižnice. Skrbniki knjižnice so osebe, ki skrbijo za vsebino knjižnice. Ti so zadolženi tudi za izposojno oziroma prevzem povračila izposojene vsebine.

2.4.3 Zgodovina izposoje

Zgodovino izposoje lahko gledamo na dva različna načina, v odvisnosti od skrbniških pravic. Če imamo skrbniško pravico, lahko pogledamo zgodovino dogodkov za vsak izdelek posebej in prav tako tudi pregled trenutno izposojene vsebine. Na spodnji sliki (Slika 2.7) lahko vidimo primer podrobnega opisa periodike, njeno vsebino in zgodovino, ki se imenuje »Karton«.



Slika 2.7: Primer zgodovine izposoje posameznih izdelkov

V primeru, da uporabnik nima skrbniških pravic, vsebina »Kartona« ne bo prikazana na spletni strani. Navaden uporabnik ima pravico pregleda le svoje lastne zgodovine izposoje, skupaj z opisom obdobja izposoje.

2.4.4 Skrbništvo nad aplikacijo

Skrbnik knjižnice ima nalogo spremljanja vsebine knjižnice. Vedno mora imeti ažuren seznam vseh knjig, periodike in medijev. Upravlja lahko tudi s šifranti založnikov, avtorjev in mest izdaje. Ko v knjižnico pride nova vsebina, jo skrbnik aplikacije doda na seznam ter določi potreben status in trenutno stanje vsebine. Za vsako vsebino imamo podatek »status«, ki nam govori ali je vsebina na zalogi, na popravilu, izgubljena ali uničena. Za vsako vsebino imamo poleg statusov tudi informacijo o dostopnosti vsebine, ki nam pove, če si lahko vsebino izposodimo za domov ali je dostopna le na društvu pod spremstvom skrbnika knjižnice. Pomemben je tudi podatek »Javno«, ki določa ali se na javnem spletnem obrazcu lahko prikaže ta vsebina ali ne. To je le redek primer vsebine, ki jo želimo imeti na seznamu, ki ga vidijo le v aplikacijo prijavljene osebe.

Skrbniki aplikacije imajo tudi zelo pomembno nalogo izposojanja in prevzema povračila izposojene vsebine. Take spremembe vodijo v aplikaciji »Knjižnica«. Za vsako vsebino se točno ve, kje se nahaja, kdaj je bila izposojena oziroma vrnjena v knjižnico. Spremembe ostajajo za vedno vpisane v bazi. Pregled take zgodovine imenujemo »Karton«. Vsaka vsebina v knjižnici ima svoj karton. Karton ima zelo pomembno vlogo – z njim lahko izsledimo razlog, če se kdaj katera vsebina poškoduje ali izgubi. Tudi za delovanje odločitvenega modula ima karton vsebine zelo pomembno vlogo, saj bomo iz njega izvedeli, koliko je vsebina zaželeno in kakšne ocene so ji priredili uporabniki.

2.4.5 Arhitektura aplikacije

V prilogi se nahaja podatkovni modul knjižnice, v katerem vidimo, da aplikacija deluje na dvanajstih različnih tabelah. Ker je v polju »Avtorji« lahko več avtorjev za eno vsebino, je bil za lažji pregled in iskanje avtorjev zgrajen pogled »KN_AVTOR_KOMB«. Ker so lahko uporabniki samo člani društva, je za potrebe delovanja aplikacije zgrajen tudi pregled članov »KN_ADRESAR«. Iz takega pregleda lahko izvemo samo osnovne informacije o članu, za katerega nam je vodstvo društva dovolilo pregled.

Za potrebe prijave v knjižnico imamo dve funkciji. Funkcija »GEN_SHA1« nam pomaga geslo zakodirati z algoritmom SHA1. Takšno geslo potem primerjamo, če je podobno v naši podatkovni bazi. Druga funkcija »KN_AUTH« preveri, če je pravilno uporabniško ime in geslo enako ustreznim podatkom v podatkovni bazi. Poleg uporabniškega imena in gesla

preverimo tudi pravico pregleda aplikacije. V primeru, da ima pravico pregleda kot navaden uporabnik, mu prikažemo vso vsebino, ki je primerna za takega uporabnika. V drugem primeru, ko je uporabnik skrbnik knjižnice, mu spletna stran omogoči prikaz celotne vsebine in možnosti spreminjanja.

Za potrebe prikaza stanja zapisa smo na bazo dodali tri funkcije, ki vračajo status izposojenega: resnično ali neresnično. Funkcije posegajo v karton za posamezen izdelek in preverijo njegovo stanje. V primeru, da njegovo zadnje stanje v polju »datum vrnjeno« nima vrednosti, pomeni, da je izdelek izposojen in funkcija vrne neresnično. Resnica bi pomenila, da je izdelek na zalogi in ga lahko v tem trenutku izposodimo. V drugem primeru, ko funkcija vrne neresnico, je izdelek že izposojen, poškodovan ali izbrisan s seznama knjižnice.

Znotraj sklopa aplikacije sodijo tudi sekvence, ki jih uporabljamo za vstavljanje novih primarnih ključev v tabele. Take sekvence največ uporabljajo prožilci, ki ob dogodkih vstavljanja vsebine v tabelo avtomatično ustvarijo ključe in ostale privzete vrednosti. V primeru akcije spremembe zapisa se spremenijo datum spreminjanja in tisti uporabnik, ki je zadnji izvedel akcijo spremembe.

Za pospešitev sistema se v bazi nahajajo tudi indeksi, ki so načeloma primarni ključi. Za indekse iskanja se načeloma izberejo vrednosti, ki se v bazi najpogosteje iščejo. S spreminjanjem sistema se v večini primerov spremenijo tudi indeksi. S podrobnimi analizami sistema najpogosteje pridemo do najboljših rešitev za določen podatkovni modul.

2.5 Priporočila

Trenutno ima knjižnica zelo malo število izposoj knjig in imajo člani veliko pripomb glede načina izbire literature. Veliko časa porabijo za izbiro knjige, ki naj bi jih mogoče zanimala. Z dodatnim odločitvenim modulom bomo izboljšali število izposoj, izgubo časa iskanja knjige, periodike ali medija in v društvo morda privabili tudi kakšnega novega člana.

Veliko članov se pritožuje tudi nad tem, da naš fond ni viden na spletu ter si tako ne morejo že doma pogledati, če knjižnica ima zeleno knjigo ali ne. To je bil eden izmed glavnih razlogov, da smo se odločili našo knjižnico vključiti v sistem COBISS ter izdelali aplikacijo.

Cilj je, da uporabnik čim enostavneje in hitro pride do zelene knjige, medija ali serijske publikacije. Z aplikacijo bi članu lažje in hitreje dali informacijo, če zeleni dokument obstaja v naši knjižnici ali ne. Uporabniki pa bodo sčasoma tudi sami zmogli uporabljati aplikacijo in brez pomoči brskali po knjižničnem fondu.

Poglavje 3 Tehnična orodja

Aplikacija »Knjižnica« deluje na Oracle podatkovni bazi in v spletnem orodju Oracle Application Express. Za delovanje odločitvenega modula smo izdelali skripto, ki jo dodamo na že obstoječo Oracle podatkovno bazo. Aplikacijo »Knjižnica« smo dodelali in ji dodali dodatno spletno stran, na kateri bodo prikazana priporočila.

3.1 Tehnične specifikacije odločitvenega modula

V tem poglavju si bomo najprej ogledali tehnologije, na katerih deluje spletna aplikacija »Knjižnica«. Prikazali bomo tudi celoten podatkovni model, ki je potreben za delovanje nove odločitvene skripte. Podatkovni modul ima zelo pomembno vlogo v samem delovanju naše odločitvene skripte, saj se tako hranijo vsi zapisi o zgodovini brskanja uporabnika in prav tako tudi nastavitve, ki so potrebne za samo delovanje skripte.

3.1.1 Oracle podatkovna baza

Oracle podatkovna baza je zbirka različnih podatkov, obravnavana kot enota. Namen podatkovne baze je shranjevanje in branje medsebojno povezanih informacij. Podatkovna baza deluje kot strežnik in je ključnega pomena za reševanje problemov upravljanja z informacijami. Podatkovni strežnik lahko upravlja z veliko količino podatkov v večuporabniškem okolju. Velika prednost strežniške baze pred lokalno bazo je ta, da lahko hkrati več uporabnikov dostopa do enakih informacij. Oracle podatkovna baza nam omogoča veliko zaupnost, celovitost in razpoložljivost informacij v vsakem trenutku. Omogoča nam tudi veliko število prepletenih informacij in veliko število vpisov ter poizvedb. Podatkovna baza nam z varnostnimi shemami, ki jih uporabnik lahko sam dogradi, tudi preprečuje nepooblašcene dostope. Vgrajeno ima tudi avtomatično izdelavo varnostnih kopij in v primeru napak se zelo enostavno povrne na zadnje delujoče stanje.

Podatkovna baza je razdeljena na logično in fizično strukturo. Fizična struktura nam govori, kam fizično shranimo informacije na trdi disk in kako dostopamo do njih. Le ta skrbi za čim učinkoviteje izkoriščen prostor, kjer so zapisani podatki. Za razliko od fizične nam logična struktura govori, kako so informacije medsebojno odvisne in kaj počnemo z njimi.

Na Oracle podatkovno bazo lahko povežemo različne aplikacije, servise ali naprave in upravljamo z informacijami znotraj podatkovne baze. Po zaslugi informatikov končnim uporabnikom ni potrebno vedeti, katera je podatkovna baza v ozadju oziroma na kakšnem sistemu deluje. Cilj podatkovne baze je končnemu uporabniku zagotoviti varen in nemoten 24-urni dostop do informacij.

3.1.2 Oracle Application Express

Oracle Application Express (Oracle APEX) je Oraclovo glavno orodje za razvoj spletnih aplikacij v kombinaciji s SQL in PL/SQL jezika. Z uporabo samo spletnega brskalnika lahko namestimo in razvijamo več različnih aplikacij za stacionarni računalnik ali mobilni telefon.

APEX je brezplačno orodje, ki je podprto za vse verzije Oracle podatkovne baze. Tudi osebe, ki nimajo znanja SQL in PL/SQL jezika, lahko s pomočjo vgrajenega čarovnika na enostaven način namestijo že obstoječe aplikacije, kot so Survey Builder, Customer Tracker in P-Track (za spremljanje projektov).

Oracle Application Express je usmerjeno razvojno orodje na Oracle podatkovni bazi, ki se lahko uporablja za izdelavo več različnih podatkovnih aplikacij. V nekaj minutah lahko izdelate enostavno tabelo v večuporabniškem okolju, ki deluje kot spletna stran. V drugem primeru lahko izdelate kompleksno, večjezično aplikacijo, kot je Oracle Store. Application Express deluje znotraj Oracle podatkovne baze ter omogoča vašim aplikacijam medsebojno izmenjevanje enakih informacij, ki so ključnega delovanja za vašo največjo skupino uporabnikov. Pri razvoju aplikacije ni potrebno veliko vedeti o varnosti, saj ima APEX vgrajeno varnost na osnovnem nivoju delovanja aplikacije, kot je preverjanje spremenljivk z varnostno sejo, preverjanjem veljavnosti uporabnika in prijavno vlogo. S tako varnim okoljem in nemotenim delovanjem uporabnikom ni potrebno skrbeti za varnost, zato lahko svojo pozornost usmerijo na poslovne cilje.

3.1.3 PL/SQL programski jezik

PL/SQL jezik se je pojavil zaradi potrebe po kompleksnih rešitvah SQL stavkov. Na splošno imajo posamezni SQL stavki zelo malo vpliva en na drugega. Zaradi premajhnega vpliva je omejeno delovanje programa, v katerem želimo zgraditi blok kode in, ki bo vplival na delovanje programa v odvisnosti od podatkov in uporabnikov. Ker bi radi zgradili aplikacije s pomočjo SQL stavkov, jih je potrebno medsebojno povezati s standardnim programskim jezikom, kot je C, oz. povečati funkcionalnost in zgraditi nov programski jezik.

PL/SQL je torej jezik za razvoj aplikacij, prevzet je iz preprostega SQL-a in podpira opcije standardnega jezika in dodatne značilnosti:

- Struktura bloka kode,
- Pogojni stavki in ponovitvene zanke,
- Spremenljivke, konstante in tipi,
- Strukturirani zapisi,
- Omogočeno upravljanje z napakami.

Še ena pomembna dodatna značilnost za PL/SQL jezik je, da omogoča shranjevanje izdelane kode direktno v podatkovno bazo. Taka značilnost omogoča neomejenemu številu aplikacij ali uporabnikom dostop do enake funkcije ali procedure. To pomeni, da lahko več uporabnikov istočasno uporablja enako kopijo kode, kar je zelo uporabno za spletne aplikacije. V primeru ene spremembe te kode pride do enake spremembe za vse uporabnike. PL/SQL jezik nam omogoča tudi izdelavo različnih prožilcev nad podatkovnimi zapisi in tako olajša upravljanje z informacijami.

Za razliko od SQL-a, PL/SQL ni standardno industrijski jezik, ampak je izključno produkt Oracle korporacije.

Zaradi boljše prilagodljivosti je PL/SQL koda programsko prevedena pred zagonom programa. Zaradi tega se ne morete sklicevati na objekte, ki še ne obstajajo v podatkovni bazi in zato je edini del SQL-a, ki ne vsebuje PL/SQL, DDL (Data Definition Language). Primer DDL stavka bi bil »CREATE TABLE«, ki ustvari podatkovno tabelo in vse njene objekte v podatkovni bazi. V primeru, da potrebujemo DDL ob zagonu kode, lahko uporabimo že zgrajen podatkovni paket DBMS_SQL, ki je del podatkovnega strežnika.

3.2 Podatkovni model

Za potrebe delovanja priporočila vsebine uporabniku uporabljamo različne podatkovne objekte, ki nam omogočajo obdelavo podatkov, potrebnih za izvedbo skripte. V nadaljevanju bomo prikazali vse objekte, ki se dogradijo na že obstoječo bazo knjižnice.

3.2.1 Podatkovne tabele

Za potrebe delovanja odločitvenega modula na že obstoječo bazo aplikacije »Knjižnica« se doda štiri različne tabele. Tabele so definirane s prvima dvema začetnima znakoma projekta. Ker gre v tem primeru za projekt »Knjižnica«, se na prvo mesto doda KN in podčrta. V naslednjem koraku se doda črka modula. Ker gre v našem primeru za »Priporočanje«, smo

dodali »P« in podčrtali. Na koncu sledi še celoten naziv tabele, ki nam na kratko pove, kaj se nahaja v tabeli.

Tabela KN_P_TIP_ISKANJA

Ime polja	Tip	Opis
<i>ID_P_TIP_ISKANJA</i>	Število	Primarni ključ tabele
<i>NAZIV</i>	Besedilo	Naziv tipa iskanja

Tabela 3.1: Seznam polj v tabeli KN_P_TIP_ISKANJA

Zgornja tabela (Tabela 3.1) KN_P_TIP_ISKANJA ima v našem primeru tri vrednosti. Vrednost »Iskanje« opisuje elemente, ki se uporabljajo v navadnem pregledu iskanja. Vrednost »Podrobno iskanje« opisuje elemente pregleda, ko se uporabnik postavi na točno izbrano vsebino. Tretja vrednost »Priporočilo« pa opisuje element, ki ga je uporabnik izbral iz pregleda predlog. Takšen element nam veliko pomeni in nas bolj podrobno usmerja v tem, kaj si uporabnik želi izposoditi.

Tabela KN_P_ISKANJE

Ime polja	Tip	Opis
<i>ID_P_ISKANJE</i>	Število	Primarni ključ tabele
<i>IP</i>	Besedilo	Vrednost, od koder prihaja vprašanje za ogled
<i>SEJA</i>	Besedilo	Uporabnikova seja
<i>NAZIV</i>	Besedilo	Naziv tipa iskanja
<i>ID_P_TIP_ISKANJA</i>	Število	Polje, povezano s šifrantom tipa iskanja
<i>VREDNOST_ISKANJA</i>	Besedilo	Vrednost, s katero je uporabnik izvedel iskanje
<i>ID_ADRESAR</i>	Število	Vrednost uporabnika, če je uporabnik prijavljen
<i>ID_TIP_ZAPISA</i>	Število	Vrednost, ki nam pove, kaj je vsebina iskanja
<i>DATUM</i>	Časovni zapis	Datum in čas iskanja

Tabela 3.2: Seznam polj v tabeli KN_P_ISKANJE

Zgornja tabela (Tabela 3.2) KN_P_ISKANJE nam sproti shranjuje vsa iskanja, ki se dogajajo znotraj aplikacije. S spremljanjem zgodovine iskanja bomo lahko izvedeli, kakšna tematika uporabnika bolj zanima in se tako tudi lažje odločili, kaj uporabniku priporočati.

Tabela KN_P_PRIPOROCILO

Ime polja	Tip	Opis
<i>ID_ADRESAR</i>	Število	Uporabnik, ki mu prikazujemo priporočilo
<i>ID_KNJIZNICA</i>	Število	Vsebina, ki jo priporočamo uporabniku

Tabela 3.3: Seznam polj v tabeli KN_P_PRIPOROCILO

Zgornja tabela (Tabela 3.3) KN_P_PRIPOROCILO se uporabi, ko se uporabnik prijavi v aplikacijo. Takrat se izvede skripta priporočila in napolni tabelo. Tabela nam pospeši delovanje aplikacije v primeru, ko ni potrebno novo računanje priporočila, ampak samo prevzamemo trenutne vrednosti tabele.

Tabela KN_P_NASTAVITVE

Ime polja	Tip	Opis
<i>ID_P_NASTAVITVE</i>	Število	Primarni ključ, po katerem ločimo nastavitve
<i>VREDNOST</i>	Besedilo	Vsebina, ki se uporabi v nastavitvi
<i>OPIS</i>	Besedilo	Podroben opis nastavitve

Tabela 3.4: Seznam polj v tabeli KN_P_NASTAVITVE

Zgornjo tabelo (Tabela 3.4) KN_P_NASTAVITVE uporabljamo kot predlogo, v kateri nastavljammo različne vrednosti za delovanje priporočilne skripte. Tabela ima zelo pomembno vlogo pri samem delovanju aplikacije, saj s spreminjanjem vrednosti vplivamo na samo delovanje aplikacije.

Tabela KN_KARTON

Ime polja	Tip	Opis
<i>OCENA</i>	Število	Ocena, ki jo poda uporabnik, ko vrne vsebino

Tabela 3.5: Dodatno polje v tabeli KN_KARTON

V že obstoječo tabelo KN_KARTON v podatkovni bazi se je dodalo polje OCENA (Tabela 3.5), ki jo poda uporabnik, ko vrne vsebino v knjižnico. Ocena je naravno število od ena do pet. V primeru, ko uporabnik ne oceni knjige, polje ostane prazno.

3.2.2 Pogledi

Pogled uporabljamo v primeru, ko pogosto posegamo v bazo za točno določene parametre v različnih tabelah, ki so medsebojno vsebinsko povezane. V našem primeru pogosto posegamo v bazo – povprečno uporabnikovo oceno vsebin želimo računati vsakič, ko se uporabnik prijavi v aplikacijo in ko je le-to potrebno za delovanje priporočilne skripte. V ta namen smo izdelali pogled z nazivom »KN_P_UPORABNIK_KN_POVPRECJE« (Tabela 3.6), ki nam z enostavnim klikom hitro izračuna povprečno oceno vsebin glede na uporabnika.

Pregled KN_P_UPORABNIK_POVPRECJE

Ime polja	Tip	Opis
ID_ADRESAR	Število	Uporabnik, za katerega računamo povprečno oceno
OCENA	Število	Povprečna ocena vseh izdelkov glede na uporabnika

Tabela 3.6: Seznam polj v pregledu KN_P_UPORABNIK_POVPRECJE

3.2.3 Paket

Za potrebe delovanja priporočilne skripte in enostavnosti pregleda, kot tudi spreminjanja baznih objektov, je bil zgrajen paket z nazivom »PRIPOROCILNI_MODUL«. Seznam vsebine in kratek opis paketa si lahko ogledate v spodnjih tabelah (Tabela 3.7 in Tabela 3.8).

Naziv funkcije	Kratek opis
VRNI_IP	Prevzamemo vrednost IP uporabnika
VRNI_ADRESAR_ID	Za uporabniško ime izvemo njegov ID
ALI_JE_KNJIGA_IZPOSOJENA	Za določen izdelek izvemo, ali je na zalogi
ALI_JE_AVTOR_V_NAZIVU	Preverimo, če iskanje vsebuje naziv avtorja
VRNI_MAKS_TIP_ISKANJA	Določimo tip vsebine, če že ni določen
VRNI_TIP_ZAPISA_ID_PODROBNO	Izvemo za podrobno brskanje, tip vsebine
VRNI_RAZMERJE_ISCI_KARTON	Prevzamemo trenutno razmerje v kartonu
VRNI_RAZMERJE_PODROBEN_PREGLED	Prevzamemo trenutno razmerje podrobnega pregleda

<i>VRNI_RAZMERJE_PRIPOROCILO</i>	Prevzamemo trenutno razmerje priporočila
<i>VRNI_ST_PRIKAZA_ZAPISOV</i>	Iz nastavitve preberemo št. zapisov prikaza
<i>VRNI_MEJO_PODOBNOSTI</i>	Iz baze izvemo št. za podobnost uporabnikov
<i>VRNI_ST_ZAPISOV_ISKANJA</i>	Izvemo število iskanja uporabnika
<i>VRNI_ST_ZAPISOV_KARTON</i>	Izvemo število izposoj uporabnika
<i>VRNI_TIP_PRIKAZA_POVPRECJE</i>	Izvemo število prikaza posamezno po tipu
<i>VRNI_TIP_PRIKAZA_POVPRECJE_VSI</i>	Izvemo skupno povprečje
<i>VRNI_TIP_PRIPOROCILA</i>	Izvemo število priporočil za posamezen tip
<i>VRNI_POVPRECJE_OCENE</i>	Izvemo povprečno oceno izdelka
<i>VRNI_NAKLJUCNO_KNJIZNICA_PODROBNO</i>	V primeru prvega obiska naključen prikaz
<i>VRNI_NAKLJUCNO_ID_ADRESAR</i>	V primeru prvega obiska naključen prikaz

Tabela 3.7: Seznam funkcij v paketu PRIPOROCILNI_MODUL

Naziv procedure	Opis
<i>PRIPOROCAJ</i>	Glavna procedura za izvajanje celotnega priporočila

Tabela 3.8: Procedura v paketu PRIPOROCILNI_MODUL

Poglavje 4 Opis delovanja odločitvenega modula

V tem poglavju bomo obdelali potek delovanja odločitve priporočila od začetka delovanja skripte do samega prikaza odločitve na spletni strani. Ogledali si bomo tudi seznam vseh privzetih vrednosti, ki so potrebne za delovanje skripte. Diagram poteka odločitve bomo razdelili v tri različne celote in na kratko opisali, kaj se dogaja v posameznem procesu odločitve. Podrobnosti delovanja skripte bomo prikazali v naslednjem poglavju.

4.1 Nastavljeni osnovni statični parametri

Za potrebe delovanja skripte je bila dograjena tabela KN_P_NASTAVITVE, ki vsebuje vse privzete vrednosti dobljene z začetnim testiranjem priporočilnega modula, ki so potrebne za delovanje priporočila. Trenutne vrednosti v tabeli lahko spreminja le skrbnik sistema. Potrebe po spreminjanju privzetih vrednosti pridejo z večjo uporabo samega sistema. Primer spremembe bi lahko bil, da ko se poveča število uporabnikov, lahko povečamo vrednost za mejo podobnosti uporabnikov. Zaradi lepšega pregleda vseh vrednosti so vsi parametri naštet v spodnji tabeli. V nadaljevanju bomo podrobno obdelali vsako vrednost in njen vpliv na delovanje odločitvene skripte.

ID	Vrednost	Opis
1	10	Število vrstic za prikaz uporabniku
2	10	Razmerje zapisa kartona in zapisa iskanja
3	5	Razmerje zapisa podrobnega in navadnega iskanja
4	5	Razmerje zapisa priporočila in podrobnega iskanja
5	0,4	Utež meje podobnosti uporabnikov
6	10	Potrebno število zapisov iskanja
7	5	Potrebno število zapisov kartona izposoje

Tabela 4.1: Seznam vseh parametrov v nastavitvah

4.1.1 Število vrstic za prikaz uporabniku

Parameter s ključem ena nam govori, koliko vrstic v priporočilu prikažemo uporabniku. Ker bomo v bližnji prihodnosti dodali v podatkovno bazo še slike za prikaz izdelka, smo se odločili

za prikaz le desetih izdelkov. Prikaz več kot desetih izdelkov s slikami bi namreč lahko bil zelo moteč za uporabnika. Cilj ni uporabnika obremenjevati z izbiro, ampak na enostaven in ne moteč način prikazati vsebine, ki ga mogoče zanimajo in s tem povečati število obiskov in izposoj vsebin iz knjižnice. Če se skrbnik odloči zmanjšati število prikazov ali pa ga povečati v primeru želje uporabnikov, lahko to naredi s spremembo tega parametra in s tem takoj vpliva na delovanje same aplikacije.

4.1.2 Razmerje zapisa kartona in zapisa iskanja

Parameter s ključem dve nam govori o pomembnosti razmerja kartona in zapisa iskanja. Ker je zgodovina izposoje pomembnejša od samega zapisa iskanja, smo se odločili za privzeto vrednost deset. Karton izposoje nam govori več o samem uporabniku, kaj si je izposodil in kakšno oceno je podal za posamezen izdelek. Zapis, ki ga hranimo v tabeli iskanja, nam pove, kaj je uporabnik iskal, za čem je brskal. Takšen zapis nam zelo malo pove o uporabniku, saj vsi uporabniki radi na hitro pregledajo vsebino celotne knjižnice. Zapis iskanja je pravilen le, ko ima uporabnik veliko število iskanj. V tem primeru lahko z analizo zapisov pridobimo podatek, ki nam govori, katera vsebina ga nekoliko bolj zanima. Torej, naše razmerje deset nam poda informacije za deset izposojenih knjig. Naš uporabnik mora imeti sto zapisov iskanja, da bi bil zapis priporočila enakovreden. Ker je naša aplikacija šele začela z delovanjem, je zelo pomembno, da ni prevelike razlike, saj imamo v bazi zelo malo število izposoj in iskanja. Ko bo aplikacija delovala nekaj let in bo število izposoj v podatkovni bazi naraslo, bo smiselno povečati število pomembnosti zapisov kartona nad zapisom iskanja. V takem primeru se bo izboljšal sam prikaz priporočila za posameznega uporabnika.

4.1.3 Razmerje zapisa podrobnega in navadnega iskanja

Parameter s ključem tri nam govori o pomembnosti razmerja podrobnega in navadnega iskanja uporabnika. Ker je zelo pomembna razlika med zapisi, smo se odločili, da je podroben zapis iskanja petkrat bolj vreden kot zapis navadnega iskanja. Tako razmerje nam tudi govori, da je v našem primeru podroben zapis iskanja dvakrat manj vreden kot zapis kartona izposoje. Kot je razvidno iz samega razmerja, je podatek podrobnega iskanja zelo pomemben, saj do takega zapisa pride, ko uporabnik klikne na posamezen zapis. V primeru takega ogleda izvemo, da uporabnika taka informacija dovolj zanima, da si je vzel čas ogledati vsebino posameznega izdelka. Tak izdelek ima v naši skripti veliko vlogo iskanja podobnih zapisov z večjo oceno kot sam zapis. S povečanjem števila navadnega in posameznega brskanja v podatkovni bazi bo smiselno, tudi z analizo zapisov, po nekem času delovanja aplikacije spremeniti samo razmerje med tema dvema zapisoma. S spremembo razmerja med zapisi

bomo tudi vplivali na samo delovanje aplikacije in s tem tudi izboljšali delovanje prikaza priporočilnega modula.

4.1.4 Razmerje zapisa priporočila in podrobnega iskanja

Parameter s ključem štiri nam govori o pomembnosti razmerja med priporočilnim in podrobnim zapisom iskanja. V našem primeru smo se odločili, da je zapis priporočila petkrat bolj vreden kot zapis podrobnega iskanja. Tako razmerje nam tudi govori, da je en priporočilni zapis petindvajsetkrat bolj vreden kot en zapis navadnega iskanja. Taki zapisi so zelo pomembni za izdelavo pravega priporočila, saj nam govorijo, kaj je bilo priporočeno in kaj od takega priporočila se je uporabnik odločil podrobneje ogledati. Zapis priporočila nam govori, kakšen tip zapisov ga zanima, saj podroben ogled pomeni, da točno vemo za kakšno vsebino gre, kateri je njen tip ter povprečna ocena uporabnikov s podobno vsebino. S tem lahko prikažemo zapise z večjo pomembnostjo in tako zgradimo boljši priporočilni sistem. Po določenem času uporabe aplikacije se bo predvideno večalo tudi število ogledov priporočila in takrat (torej po nekem času) bo smiselno izvesti analizo različnih vrst zapisov in v aplikacijo postaviti nova razmerja. Z izvedbo analize in spremembami razmerja vplivamo neposredno na delovanje aplikacije in s tem izboljšamo tudi samo delovanje odločitvenega modula.

4.1.5 Utež meje podobnosti uporabnikov

Parameter s ključem pet nam govori, katero najmanjšo vrednost mora imeti izračun podobnosti uporabnikov, da lahko trdimo, da se uporabnika ujemata. Več o računanju podobnosti uporabnika sledi v naslednjem poglavju, kjer bomo obdelali tudi problematiko povprečne ocene posameznega uporabnika, ki zelo vpliva na odločitveno skripto. V našem primeru smo se odločili za manjšo mejo podobnosti uporabnikov z vrednostjo 40% oz. 0,4 v podatkovni bazi. Za manjšo vrednost ujemanja uporabnikov smo se odločili, ker v bazi nimamo veliko zapisov v kartonu izposoj. S številom izposoj bo raslo tudi število različnih tipov uporabnikov. V takem primeru se nam splača povečati utež meje podobnosti uporabnikov. Z večjo mejo pridobimo približno enake profile uporabnikov in s tem izboljšamo delovanje samega odločitvenega modula.

4.1.6 Potrebno število zapisov iskanja

Parameter s ključem šest nam govori, koliko zapisov je potrebno, da upoštevamo zapise iskanja v odločitvi števila tipov zapisov za prikaz na priporočilni spletni strani. Ker skripta najprej sestavi število tipov prikaza vsebine, je zelo pomembno, od kod izvemo uporabnikovo zgodovino. Ker je naša baza komaj začela z delovanjem, imamo zelo malo število zapisov, po

katerih bi izdelali naše odločitve. Za začetek delovanja naše skripte smo nastavili zelo majhno vrednost, in sicer število deset. V takem primeru, ko je samo deset zapisov, pomeni, da je teh deset zapisov enakovredno kot en zapis v kartonu izposoje. Po določenem času, ko bo zelo naraslo število izposoj, bi bilo logično povečati tudi parameter, ki nam govori, koliko zapisov iskanja je potrebno, da vpliva na našo odločitev. Takšni zapisi so namreč zelo različni in povedo zelo malo o uporabniku v primerjavi z zapisi iz kartona izposoje.

4.1.7 Potrebno število zapisov kartona izposoje

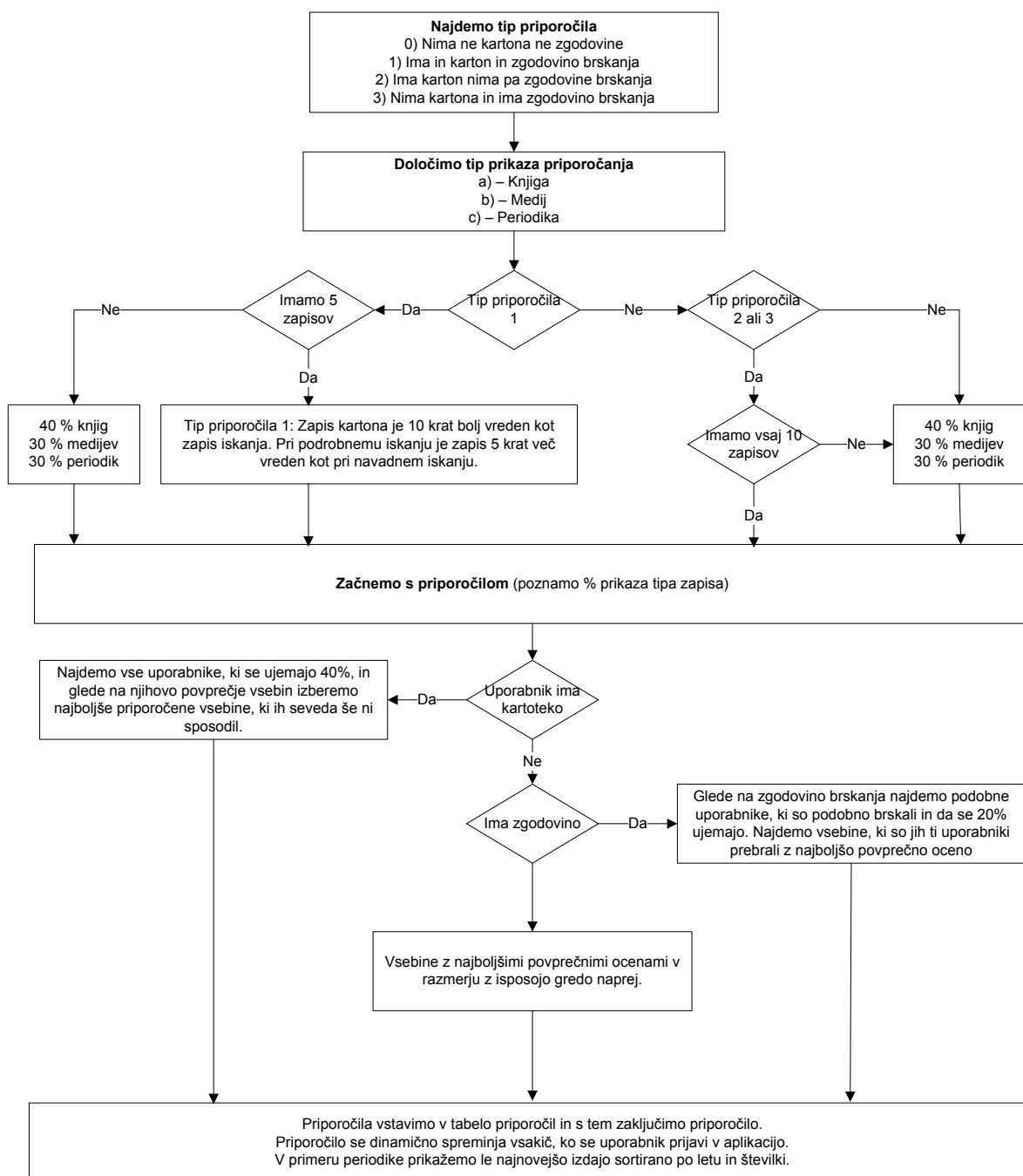
Parameter s ključem sedem nam govori najmanj koliko zapisov je potrebnih za izvedbo odločitvene skripte. Za privzeto vrednost pet smo se odločili, ker je naš sistem komaj v začetnem stanju delovanja. V bazi imamo zelo malo zapisov in iz takih zapisov naj bi sestavili najboljšo možno odločitev. En zapis kartona izposoje je vreden kot deset zapisov iskanja in v tem primeru je potrebno manjše število za enako dobro odločitev. Parameter s številom zapisov kartona izposoje je eden od redkih, ki se ne bo spreminjal s spremembo števila izposoj. V odločitvenem modulu je naš cilj imeti enako dobra priporočila tako za stalne kot za nove uporabnike. Ko se v aplikacijo prijavi novi uporabnik, nimamo veliko zapisov o njegovem profilu. V takem primeru je pomemben katerikoli podatek, ki bi nam pomagal v izdelavi priporočila. V primeru, ko povečamo takšen parameter, ne povečamo kvalitete za že obstoječe uporabnike, temveč vpliv kvalitete priporočila le na nove uporabnike.

4.1.8 Privzet osnovni prikaz priporočila

Privzet osnovni prikaz priporočila ni v parametru, ampak nekako logična nastavitve priporočila v primeru, da ne vemo nič o uporabniku. V primeru, ko uporabnik nima niti zgodovine brskanja in zgodovine izposojanja niti dovolj velikega števila iz parametra pet in šest, mu prikažemo privzeti osnovni prikaz priporočila. Najpomembnejša odločitev, po kateri potem nadaljujemo, je odločitev, koliko različnih tipov prikaza posredovati uporabniku. Privzeto vrednost sestavlja enostavna formula, ki sešteje število različnih tipov vsebine in deli s številom iz parametra ena. V našem primeru imamo tri tipe vsebine, in sicer knjiga, periodika in medij. V parametru ena imamo vrednost števila prikazov zapisov na spletni strani, in sicer deset. Ko deset delimo s tri, dobimo 3,33. Ker to ni celo število, zaokrožimo navzdol in dobimo tri. Za vsak tip vsebine bomo prikazali tri zapise, vendar pa se to ne ujema z našim seštevkom. Potrebno je dodati še en dodaten zapis vsebine. Takšen zapis izberemo s seštevkom celotne vsebine po tipu zapisov. Ker ima knjižnica trenutno največ knjig, bo privzeta vrednost za uporabnika 4 knjige, 3 periodike in 3 medije. O tem, kaj bomo od posameznih tipov vsebine priporočali, pa sledi bolj podroben opis načina izbire priporočila v naslednjem poglavju.

4.2 Diagram poteka priporočila

Diagram poteka nam bo na hitro razložil, kako poteka proces odločitve v odvisnosti od različnih tipov uporabnikov. Opis diagrama je bil razdeljen v smiselne podenote delovanja skripte. Zgled celotnega procesa si lahko ogledamo v diagramu poteka odločitve na spodnji sliki (Slika 4.1).



Slika 4.1: Diagram poteka procesa priporočila

4.2.1 Določanje števila in tipa prikaza vsebine

Potek delovanja skripte prične z izbiro tipa uporabnika. Uporabnike najprej ločimo na prijavljene in neprijavljene. Za neprijavljene uporabnike imamo manj informacij, saj ne vemo, kdo so in zato gradimo poročilo glede na njihovo sprotno brskanje in večinoma po privzeti vsebini, enako kot za nove uporabnike. Za uporabnike, ki so prijavljeni v sistem, imamo več informacij, in sicer celotno zgodovino brskanja uporabnika kot tudi celoten seznam vseh izposojenih vsebin. Za prijavljenega uporabnika priporočilo vedno izgleda boljše, ker že od prej vemo, kaj ga zanima.

Ko določimo tip uporabnika, skripta nadaljuje z iskanjem zgodovine brskanja in izposojanja vsebine. Skripta izve točno število in zgradi enega od štirih različnih profilov uporabnika, in sicer uporabnika brez zgodovine brskanja in zgodovine izposojanja; uporabnika, ki ima tako zgodovino brskanja kot zgodovino izposojanja; in še zadnja dva primera, ko ima zgodovino brskanja, nima pa zgodovine izposojanja ter obratno.

Skripta loči proces v odvisnosti od profila uporabnika in začne z računanjem števila prikaza različnih tipov vsebin. Za izračun števila prikaza upošteva podane parametre iz tabele z nastavitvami parametrov. V primeru, da skripta ne najde nobenega zapisa oz. če ima premalo zapisov zgodovine brskanja in zgodovine izposoje, nam izračuna in prikaže kar osnovno privzeto vrednost. Enako se zgodi, ko pride nov uporabnik oz. neprijavljen uporabnik, ki ima zelo malo zgodovine brskanja – v našem primeru manj kot deset zapisov.

Ko nam skripta določi točno število prikaza vsebine po tipu vsebine za izbranega uporabnika, nadaljujemo s procesom, ki nam točno določi, katere izdelke prikazati znotraj posameznih tipov vsebin.

4.2.2 Izvedba odločitve priporočila

Potek odločitve priporočila se prične s preverjanjem števila zapisov v kartonu izposojanja. V primeru, da ima uporabnik več kot pet zapisov v kartonu, začnemo z iskanjem podobnih uporabnikov. Če najdemo enega ali več podobnih uporabnikov, ki so podobni in, če je to več kot je meja podobnosti priporočila, potem naredimo še izračun povprečne uporabnikove ocene in v kombinaciji z ocenami knjige poiščemo, katero vsebino je več podobnih uporabnikov bolje ocenilo. Takšna vsebina ima v knjižnici najboljšo priporočilo in še posebej za uporabnika, ki ima več podobnih uporabnikov. Ko ne dobimo dovolj informacij oziroma premalo število vsebine, ki je potrebna za odločitev, se lotimo poseganja po metodi »najbolj izposojeno v sorazmerju z oceno izposoj«.

Če uporabnik nima kartona izposoje oziroma ima za naš primer manj kot pet zapisov v kartonu izposoje, naša skripta nadaljuje na preverjanje druge najboljše možnosti – preverjanje z zapisi iz zgodovine brskanja, če le obstajajo opisi. Če ima uporabnik zapise brskanja, so le-ti prioritetni s podrobnim in priporočilnim brskanjem. Take zapise poskusimo primerjati z najbolj podobnim uporabnikom v kartonu zgodovine. Če slučajno obstaja podoben uporabnik, ki je podoben več ali enako kot je meja podobnosti uporabnika, prevzamemo njegove dobre ocenjene vsebine skupaj z njegovo povprečno oceno. Ko uporabnik z zgodovino v kartonu nima podobnega uporabnika, preverimo, če obstaja podoben uporabnik po zgodovini iskanja. V takem primeru prevzamemo od drugega uporabnika kakšne zapise, ki jih je le-ta dodatno podrobno pregledal. Za takšne zapise sestavimo povprečne ocene od skupne izposoje v sorazmerju s številom izposoj in nato te zapise priporočamo.

Najslabša možnost je, ko uporabnik nima ne kartona iskanja ne zgodovine brskanja oziroma, v našem primeru, več kot deset zapisov zgodovine brskanja. V takem primeru skripta začne z osnovno predlogo. Skripta pregleda vse vsebine, ki so bile izposojene in ocenjene, izračuna povprečno oceno uporabnika ter odvzame od dane ocene, tako vrednost doda v izračun najboljše ocenjenih vsebin.

Po končani izbiri zapisov, ki jih prikažemo uporabniku, vstavimo zapise v podatkovno tabelo KN_P_PRIPOROCILO, ki se nahaja v podatkovni bazi. Zapise vstavimo v tabelo zaradi števila poizvedb. Uporabniku lahko večkrat prikažemo iste podatke brez novega računanja celotnega postopka. V primeru, da uporabnik brska po določenem številu zapisov in si izposodi različno število vsebin, lahko na novo izračunamo nove vrednosti vsebine prikaza. S tabelo pridobimo izboljšan čas prikaza in hitrosti podatkovne baze.

4.2.3 Prikaz odločitve

Ob obisku uporabnika na spletni strani v spletnem vmesniku prikažemo rezultate, ki jih zgradi skripta odločitve. Ko uporabnik prvič obiše spletno stran, še ni prijavljen in ne vemo, kdo je in kaj ga zanima. V takem primeru prikažemo zapise, ki jih skripta že ima v tabeli in s tem zmanjšamo nepotrebno delovanje skripte v podatkovni bazi. Ko se uporabnik prijavi oziroma brska za večimi različnimi vsebinami, začne skripta z delovanjem in zgradi profil uporabnika, kakor tudi vstavi zapise v tabelo s pravilnimi priporočilnimi zapisi.



Vsebina	Naziv	ISBN/ISSN	Avtorji	Založnik	Datum izdaje	Tip medija
Knjiga	Čarovnikov vajenec	9789616067621	Evald Flisar	Kulturno Umetniško Društvo	2010	-
Knjiga	Severnica nad Olimpom	9789610114567	Jelka Ovaska	Mladinska knjiga Založba	2011	-
Knjiga	Grobница	9789610110576	Kate Mosse	Mladinska knjiga Založba	2010	-
Knjiga	Harry Potter 6 - Princ mešane krvi	9789610118350	J. K. Rowling	Mladinska knjiga Založba	2014	-
Knjiga	Čudežni Feliks	9789612412685	Andrej Hieng	Modrijan Založba	2009	-
Knjiga	Proti severnemu vetru	9789612317980	Daniel Glattauer	Cankarjeva Založba, d.o.o.	2013	-
Medij	Zvitli kosmatinčki	-	Bill Kopp	Net d.o.o.	2013	-
Medij	Aktivno življenje - zdravo življenje	-	Manuela Pendl Žalek	Rotis	2004	-
Medij	Slike Pevski zbor Knežak	-	Milan Grlica	-	-	DVD
Periodika	Planinski vestnik	-	-	PZS	2014	-

Slika 4.2: Prikaz spletne strani s priporočili

Zelo pomembna postavka je izgled priporočila v spletnem vmesniku, saj če vsebina ni estetsko lepo prikazana, si je nihče ne bo ogledal ali pa celo prenehal z uporabo naše spletne strani. Zaradi tega razloga smo se v knjižnici odločili, da zgradimo dodatno spletno stran za prikaz izključno priporočil. Uporabnik, ki ga zanima priporočilo, se bo že sam prestavil na spletno stran s prikazom priporočil, z enim samim klikom na povezavo »Priporočila«. Primer spletne strani s priporočili si lahko ogledamo na zgornji sliki (Slika 4.2).

Na spletni strani s priporočili ima uporabnik možnost podrobnega ogleda priporočila. Če si bolj podrobno ogleda zapis, takšen zapis vstavimo v bazo in spet izvedemo skripto, ki bo ponovno izvedla odločitev in uporabnika usmerjala v vsebino, ki ga najbolj zanima.

V prihodnosti načrtujemo nadgradnjo priporočila s prikazom slik pred vsakim zapisom. Namen tega je izboljšava same spletne strani in bolj pregleden prikaz zapisov uporabniku. V načrtu nadgradnje je tudi t. i. košarica, v kateri bi uporabnik označil, kaj ga najbolj zanima, in si označbo vsebine shranil za izposajo. Tak podatek bo nekoč pomembno vplival na delovanje skripte, saj bo takšen zapis enakovreden zapisu izposoje.

Poglavje 5 Podroben opis funkcionalnosti priporočila

V tem poglavju bomo obdelali posamezne procese, ki imajo zelo pomembno vlogo v samem delovanju skripte. Najprej bomo obdelali posamezne razlike med zapisi, kako jih obdelamo in izvlečemo najboljšo možno informacijo iz vsakega izdelka posebej. V nadaljevanju si bomo ogledali tudi razlike med posamezni tipi uporabnikov in razlike v vplivu na delovanje skripte. Prikazali bomo tudi celoten proces iskanja števila, potrebnega za izpis posameznega tipa vsebine. V procesu je zelo pomembna tudi procedura iskanja podobnega uporabnika in prevzem njegovih dobrih ocen v sorazmerju z njegovim celotnim povprečjem. Na koncu poglavja bomo obdelali tudi iskanje povprečja za posamezen tip vsebine. Takšen proces ima zelo pomembno vlogo za nove uporabnike.

5.1.1 Razlike med zapisi

V aplikaciji ločimo več vrst zapisov, in sicer razlike med vsebino izdelkov in vsebine uporabnikove zgodovine. Obe vrsti razlik igrata pomembno vlogo v odločitveni skripti, ki določa, kaj prikazati uporabniku na spletni strani.

Vsebino izdelkov delimo na tri tipe: na knjigo, periodiko in medij. Vsem izdelkom je skupen podroben opis posameznega zapisa. Ker je v knjižnici več enakih izdelkov, lahko takšne izdelke vpišemo v knjižnico kot enak posamezen izdelek. Te izdelke ločimo po podrobnem zapisu – šifri izdelka.

V nadaljevanju bomo prikazali, katera polja vsebujeta tabeli KN_KNJIZNICA in KN_KNJIZNICA_PODROBNO ter opisali razlike med posamezni zapisi.

Ime polja	Tip	Opis
<i>ID_KNJIZNICA</i>	Število	Primarni ključ tabele
<i>ID_TIP_ZAPISA</i>	Število	Tip vsebine (knjiga, periodika ali medij)
<i>NAZIV</i>	Besedilo	Naziv vsebine
<i>ISBN</i>	Besedilo	ISBN knjige v primeru zapisa knjige
<i>ID_ZALOZNIK</i>	Število	Založnik vsebine
<i>DATUM_IZDAJE</i>	Datum	Datum izdaje vsebine
<i>ID_MESTO_IZDAJE</i>	Število	Mesto izdaje vsebine

<i>JAVNO</i>	Besedilo	Ali je zapis javen, »DA« ali »NE«
<i>AKTIVEN_ZAPIS</i>	Besedilo	Aktiven oz. neaktiven zapis; ne bo viden v aplikaciji
<i>AVTOR</i>	Besedilo	Avtor oz. seznam vseh avtorjev
<i>SIGNATURA</i>	Besedilo	Signatura za lažje iskanje
<i>OPIS</i>	Besedilo	Podroben opis vsebine

Tabela 5.1: Seznam polj v tabeli KN_KNJIZNICA

Kot je vidno iz tabele (Tabela 5.1), je največja razlika v vsebini izdelka med vsebino knjige in medija. Knjiga vsebuje polje ISBN in Signaturo – za razliko od medija, pri katerem to ni potrebno. Medij vsebuje polje »tip medija«, ki v primeru, ko gre za knjigo, ni potrebno. V primeru periodike je polje ISBN enako kot v knjigi, le da ima v spletni aplikaciji to polje pomen ISSN. Razlika med periodiko in medijem je enaka kot razlika med medijem in knjigo. Podatki javno in aktiven zapis nam govorijo, ali zapis prikazujemo na spletni strani oziroma, ali zapis sploh upoštevamo v izračunu priporočila.

Ime polja	Tip	Opis
<i>ID_KNJIZNICA_PODROBNO</i>	Število	Primarni ključ tabele
<i>ID_KNJIZNICA</i>	Število	Vezano na izdelek
<i>ID_DOSTOP</i>	Število	Lokacija, kje lahko imamo vpogled izdelek
<i>ID_STANJE</i>	Število	Trenutno stanje vsebine
<i>OPIS</i>	Besedilo	Podroben opis posameznega izdelka
<i>STEVILKA</i>	Besedilo	Številka v primeru

Tabela 5.2: Seznam polj v tabeli KN_KNJIZNICA_PODROBNO

Kot vidimo, ima tabela KN_KNJIZNICA_PODROBNO (Tabela 5.2) zelo pomembno vlogo v samem delovanju knjižnice, kot tudi v delovanju odločitvene skripte. Tabela vsebuje šifre posameznih izdelkov. Sami izdelki so razvrščeni v določeno vsebino. Tak primer bi bil več kopij enake knjige, ko so vse kopije enake, razlikujejo se le v šifri izdelka. Malo drugačen primer je za periodiko, kjer je vsa vsebina enaka, razen v številki izdane periodike in šifri izdelka. Za tak primer, ko je različna številka izdelka, smo v tabelo dodali dodatno polje »STEVILKA«.

Vrednost zapisov je enako vredna, saj uporabnik sam določi, kaj ga bolj zanima. V skripti odločitve pride v poštev le razlika v številki periodike. Logično je, da uporabnika bolj zanima nov izdelek periodike kot starejša verzija. Še posebej pride do razlike v časopisih, kot je na

primer *Delo*, saj nov izvod prinaša sveže informacije. Starejši izdelki ostajajo v knjižnici le v arhivske namene.

Ko govorimo o razlikah zapisov, je druga večja porazdelitev zapisov porazdelitev po uporabnikovi zgodovini. Vsak uporabnik knjižnice ima več različnih vrst zgodovine. Najbolj pomembna zgodovina, po kateri oblikujemo najboljše priporočilo, je zgodovina izposojene vsebine. Poleg zgodovine izposoje obstaja še zgodovina brskanja.

Zaradi velike razlike med izposajo knjige in brskanjem o njej po spletu, smo določili utež, ki je shranjena v podatkovni bazi in se lahko sproti spreminja. Za utež smo določili vrednost 10, ker je v našem primeru deset zapisov brskanja enakovredno enemu zapisu izposoje. V zgodovino izposoje vpisujemo tudi oceno, ki jo lahko poda uporabnik, ko vrne izdelek v knjižnico. Na podlagi takšne ocene skripta gradi povprečne ocene in išče knjige, ki najbolj ustrezajo posameznemu uporabniku.

Nekoliko manj pomembna zgodovina brskanja ima tudi veliko vlogo v odločitveni skripti, saj za nove uporabnike, ki si še niso ničesar izposodili, ne vemo nič. Ko nov uporabnik brez zgodovine brskanja brska po spletu, mu sproti gradimo njegov profil in primerjamo s podobnim uporabnikom. Torej je za nove uporabnike zgodovina brskanja velikega pomena v delovanju odločitvene skripte.

Zgodovina brskanja je razdeljena na tri različne tipe brskanja: navadno, podrobno in priporočilno brskanje. Vsak tip brskanja ima določen pomen v odločitveni skripti, zato so med zapisi postavljene uteži. Takšne uteži med zapisi brskanja nam govorijo o pomembnosti zapisa, saj je podroben zapis brskanja vreden več kot navaden. Kadar uporabnik brska in išče kakšno knjigo, periodiko ali medij, to imenujemo normalen tip iskanja in ga zapisujemo v bazo s tipom normalnega iskanja. Ko uporabnik klikne na določeno vsebino in si podrobno ogleda vse njene izdelke, to pomeni, da si je uporabnik vzel čas za podroben ogled vsebine in želi o njej izvedeti več. Takšen način pregleda imenujemo podrobno iskanje in ga zapisujemo v podatkovno zgodovino brskanja s tipom podrobnega brskanja. Zadnji tip, priporočilo brskanja, se zgodi, ko si uporabnik na priporočilni strani podrobneje ogleda vsebino. Takšen tip brskanja nam o uporabniku pove največ, saj je skripta že pripravila, kar ga zanima.

Med samimi tipi brskanja je očitna razlika, zato so tudi podane uteži med različnimi tipi brskanja. V našem primeru je podroben tip brskanja petkrat bolj vreden od navadnega zapisa brskanja. Priporočilni tip brskanja ima enako razmerje s podrobnim zapisom kot podrobni zapis in navaden zapis brskanja. Vse uteži se nahajajo v podatkovni bazi, in sicer v tabeli z nastavitvami. Prednost takih zapisov je, da nam je omogočena zamenjava posamezne uteži. S spremembo uteži neposredno vplivamo na delovanje odločitvene skripte. Po določenem času

delovanja je smiselno izvesti analizo vseh zapisov in na podlagi rezultatov analize ponovno določiti smiselne uteži za delovanje odločitvene skripte.

5.1.2 Razlike med uporabniki

Glavna porazdelitev uporabnikov je na prijavljene in neprijavljene uporabnike v spletno aplikacijo »knjižnice«. Prijavljeni uporabniki so lahko le člani društva, saj se je potrebno včlaniti v društvo, da bi imeli pravico do izposoje vsebine v knjižnici. Ob včlanitvi se oblikuje tudi uporabniško ime in geslo uporabnika. Neprijavljen uporabnik je lahko kdorkoli na spletu, ki je obiskal našo spletno stran knjižnice. O neprijavljenih uporabnikih ne vemo veliko, saj ne vemo, kdo so, prav tako pa tudi nimamo njihove zgodovine. Odločitvena skripta se drugače obnaša glede na neprijavljene uporabnike, in sicer tako, da najprej prikaže privzete vrednosti celotne knjižnice. Skripta preveri vse zapise izposojanja in naredi povprečne ocene glede na število izposoj in povprečne ocene izdelka. Tako predlogo priporočil prikažemo neprijavljenemu uporabniku. Ko neprijavljen uporabnik začne brskati po aplikaciji, začnemo graditi profil uporabnika. Primerjati začnemo s podobnim in priporočamo več usmerjenih priporočil. Za neprijavljene uporabnike sta zelo pomembna zapisa IP in trenutna seja, saj tako ločimo zapise med uporabniki.

Uporabniki, ki so se prijavili v aplikacijo, so znani uporabniki in v večini primerov poznamo zgodovino brskanja kot tudi zgodovino izposoje, če jo le-ta ima. Za boljše upravljanje s skripto prijavljenega uporabnika ga razdelimo na eno od štiri možnosti:

- Uporabnik brez zgodovine in brez zgodovine izposoje;
- Uporabnik z zgodovino brskanja in brez zgodovine izposoje;
- Uporabnik brez zgodovine brskanja in z zgodovino izposoje;
- Uporabnik z zgodovino brskanja in z zgodovino izposoje.

Uporabnika brez zgodovine in brez zgodovine izposoje obravnavamo enako kot neprijavljenega uporabnika. V takem primeru odločitvena skripta torej generira privzete priporočilne odločitve iz vseh zapisov zgodovine izposoje ter zgradi povprečje izposojenega števila izdelkov v kombinaciji s povprečno oceno izdelka. Ko uporabnik prične brskati, se skripta preseli na drugačen tip uporabnika, in sicer na opcijo Uporabnik z zgodovino in brez zgodovine izposoje.

Uporabnika z zgodovino brskanja in brez zgodovine izposoje obravnavamo kot znanega uporabnika. Uporabnik ima zgodovino brskanja, vendar pa si še nikoli ni izposodil vsebine v knjižnici. Takemu uporabniku zgradimo profil in poiščemo podobnega člana z najbolj podobno zgodovino brskanja. Iz podobne zgodovine uporabnika izvemo, katere vsebine so

drugega člana zanimale oz. katere je večkrat prebrskal. V primeru, da je kakšno vsebino podrobno pogledal, bo imel tak zapis večjo pomembnost in bo obravnavan bolje kot navadno brskanje. Taki uporabniki so v večini primerov člani, ki so prišli obiskat knjižnico na spletno stran in, kjer si bodo mogoče pregledali, kaj bi si izposodili pri naslednjem obisku knjižnice.

Uporabnik brez zgodovine brskanja in z zgodovino izposoje je uporabnik, ki je že bil v knjižnici in si je že izposodil različne izdelke. Takšni uporabniki imajo večjo prednost od uporabnikov, ki imajo le zgodovino brskanja, saj dejansko vemo, kaj so si izposodili in, katera vsebina jih zanima. Ko takšne člane primerjamo med seboj, dobimo bolj kvalitetne podobne uporabnike. Iz takih primerov primerjanja izvemo, kaj je bilo podobnemu uporabniku všeč oziroma, če gre za več podobnih uporabnikov, se zgradi povprečje vseh izdelkov, ki imajo nadpovprečno oceno. Da bi bil izračun verodostojen, je potrebno upoštevati uporabnikovo povprečno oceno v izračunu povprečne ocene knjige. Pomembno je omeniti, da nimajo vsi uporabniki enake razlike v ocenah. Primer bi bil uporabnik, ki v večini primerov poda vsebini nadpovprečno oceno, tokrat pa, za razliko, poda slabšo oceno. V takem primeru lahko od takšnega uporabnika izvemo, da mu vsebina ni bila všeč in je ne bomo priporočali podobnemu uporabniku. V primeru, ko uporabnik podaja primerne ocene za vsako knjigo, bo njegova ocena bolj vplivala na končno odločitev skripte.

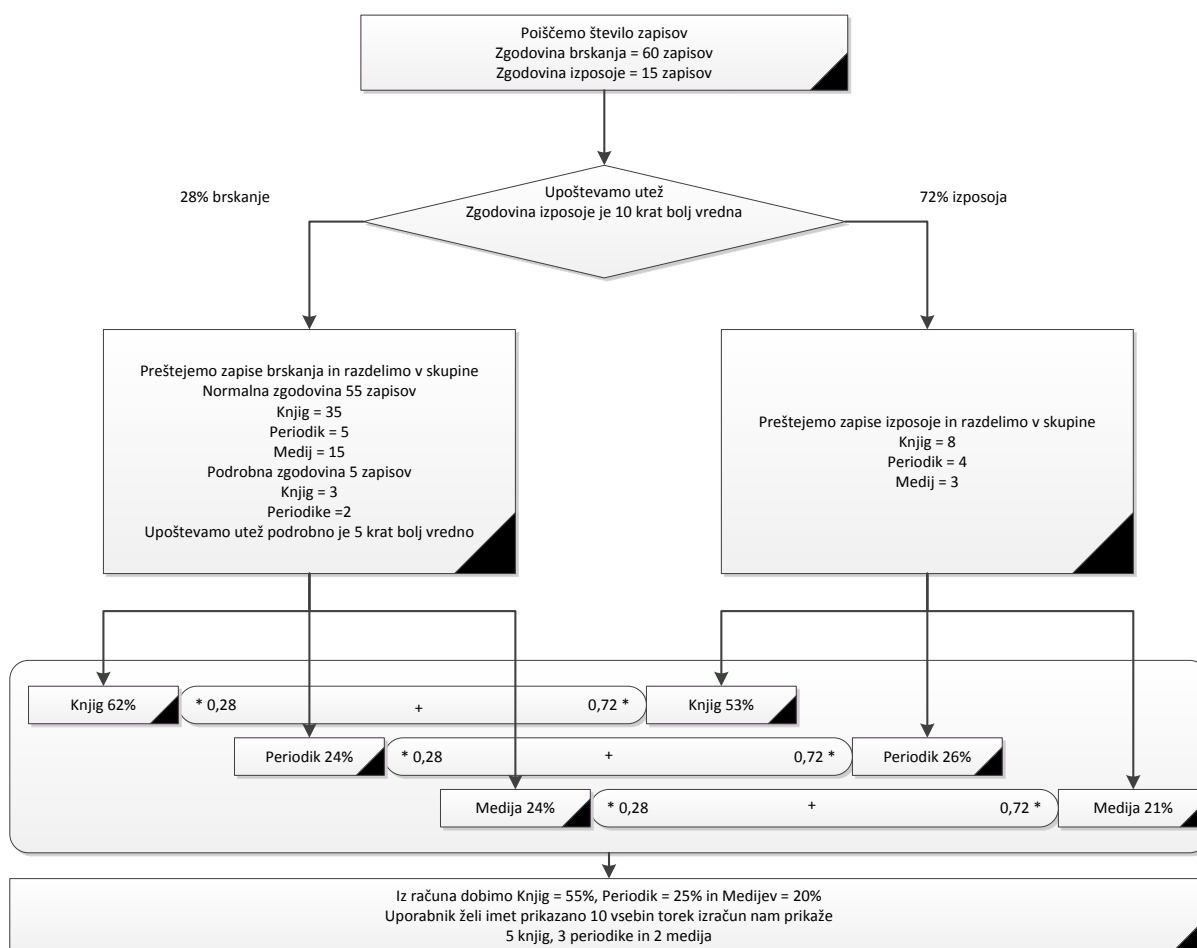
Uporabnik z zgodovino brskanja in z zgodovino izposoje je idealen uporabnik. Od takšnega uporabnika izvemo največ potrebnih informacij za izdelavo priporočila. Uporabnika lahko primerjamo z ostalimi uporabniki, ki so si že izposojali knjige in še dodatno lahko preverimo, če je katera od teh vsebin bila v zgodovini brskanja, kar je dodatno potrdilo zanimanja za neko vsebino. Od takšnih uporabnikov lahko pričakujemo, da bodo najverjetneje večkrat uporabljali priporočilne skripte, saj je tudi njihov cilj izposoditi si kakšno dobro knjigo, ki je že dobila dobro oceno s strani drugega člana. Za takšne člane je cilj tudi zgraditi čim boljšno odločitveno skripto.

5.1.3 Iskanje števil prikaza različnih tipov vsebin

Ker imamo različne tipe zapisov in različne tipe uporabnikov, je naš cilj prikazati uporabniku več različnih vsebin in ne samo eno vsebino. Primer: ko si uporabnik izposodi deset knjig, naš cilj ni priporočati samo knjige, ampak tudi ostalo gradivo iz knjižnice. Mogoče bi uporabnika utegnili zanimati tudi vsebina, kot sta periodika ali medij s slikami zadnjega nastopa v društvu.

V odločitveni skripti pridemo najprej do algoritma, ki nam prešteje vse vsebine in izračuna povprečja ter določi, koliko katere vsebine prikazati uporabniku. Skripta najprej prešteje, koliko zapisov obstaja v zgodovini brskanja in koliko zapisov zgodovine izposojanja. V

izračunu se upošteva tudi utež, ki govori kolikokrat je zgodovina izposojanja pomembnejša od zgodovine brskanja. V našem primeru (ker nimamo veliko zapisov v podatkovni bazi) je vrednost uteži določena z razmerjem deset. Ko smo določili razmerje zgodovine brskanja in zgodovine izposoje, nam skripta posebej pogleda v vsako zgodovino in izračuna razmerja med vsebinami znotraj zgodovin. Izračun nam nato posreduje nazaj in upošteva razmerja med zgodovinami ter določi končno vrednost prikaza.



Slika 5.1: Diagram poteka izbire števila vsebine za priporočanje

Na zgornji sliki (Slika 5.1) si lahko ogledamo primer, ki kaže potek izbire števila prikaza vsebine – za zgodovino brskanja število 60 in izposojene izdelke število 15. Skripta za izračun razmerja upošteva utež, ki se nahaja znotraj tabele nastavitve v podatkovni bazi. V našem primeru je ta vrednost deset, kar pomeni, da je 15 zapisov desetkrat pomembnejše. Iz tega razmerja preberemo, da skripto delimo na 28 % vrednosti zgodovine brskanja in 72 % zgodovine izposoje. V zgodovini izposojanja ni posebnih oziroma različnih tipov izposoje, tako da enostavno preštejemo vsa števila in razdelimo na povprečje izposojenih vsebin. V zgodovini brskanja pa imamo tri različne tipe zgodovin, ki jih je potrebno upoštevati pri

končnem izračunu povprečja vsebin. V zgornjem primeru je prikazana samo razlika med normalnim in podrobnim brskanjem. V izračunu je bila upoštevana utež, ki govori, da je podrobna zgodovina petkrat pomembnejša od navadne zgodovine brskanja.

Kot vidimo, je končni delež naslednji: 55 % knjig, 25 % periodike in 20 % medijev. Ker je v podatkovni bazi zapisano število prikazovanj deset, skripta preračuna in zaokroži navzdol, kar nam dá največje število prikazovanj, in sicer v tem primeru za prikaz pet knjig, tri periodike in dva medija. V primeru, da se seštevek vseh prikazov ne ujema, skripta poveča zadnjo najmanjšo vrednost za ena in s tem popravi število zapisov za prikaz.

V primeru, ko uporabnik nima ne zgodovine brskanja ne zgodovine izposojanja, nam skripta dodeli število zapisov, ki jih želimo prikazati s številom različnih vsebin. Ker je v podatkovni bazi zapisano želeno število prikazov 10 in različnih vsebin 3 nam skripta izračuna, da prikaže tri knjige, tri periodike in tri medije. Ker seštevek vseh treh rezultatov ni enak številu za prikaz, nam skripta prikaže, katera vsebina je bila največkrat izposojena in to vsebino poveča za ena. Pravilni končni rezultat privzete priporočilne skripte bodo torej štiri knjige, tri periodike in tri medije.

5.1.4 Iskanje podobnih uporabnikov

Procedura iskanja podobnih uporabnikov ima največjo vlogo v odločitveni skripti. Ta procedura nam poišče podobne uporabnike, jih primerja in izračuna ocene ter izbere, kaj priporočati uporabniku. Delovanje skripte bo najlažje prikazati na primeru in opisati celoten postopek delovanja, vključno s formulami, ki se uporabljajo v izračunu. V spodnji tabeli je podan primer za deset različnih uporabnikov.

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	UPn	r(X,n)
Ux	5	4	1	3	4	5	4	5	2	1									3,40	
U1	5	4	1	3	4	5	4	5	2	1	5	1	1		1	4	2	5	3,12	0,98
U2	4		1		4	5	3	2	2	5	5	2	5		3		5		3,54	0,28
U3	3		3	5		1	2		3	5	1		3		2	4		1	2,75	-0,57
U4	4	4	3	2	1	5	3	4	2	2	2	4	2	1	4	3	1	5	2,89	0,56
U5	5	3	2		3	4		4	1		5	2	4		1	3	2	1	2,86	0,74
U6	1	2	3	4	5	5	3	3	2	1	4	4	5	3	2	4	3	5	3,28	0,30
U7	2	1	2	3	4	5	2	5		1	5	2		1		3	2	5	2,87	0,50
U8							4		1	2	5	5	5		2	4	3	2	3,30	0,52
U9	1	2		2	1	1	1	2	1	1	2		1		3	2	3		1,64	0,03
U10	1	2	5	5	2	1	2	1	4	5	1	1		4		5	1	1	2,56	-0,92
ocena											4,16	2,36	2,31		2,11	3,74	3,1	3,86		

Tabela 5.3: Primer odločitve priporočila podobnih uporabnikov

Preden se poglobimo v potek delovanja skripte, bomo najprej razložili osnovne pojme v našem primeru. Un pomeni uporabnik z n številko; Kn pomeni ocena vsebine s številom n; Upn pomeni povprečna ocena uporabnika s številko n; Upx pomeni povprečna ocena uporabnika UX. UX je uporabnik, za katerega računamo priporočilo in ocena(n) je povprečje, ki ga izračunamo za vsako vsebino posebej.

Podpora je število, ki nam pove, koliko zapisov vsebine se mora ujemati oziroma biti enako, da uporabnika upoštevamo pri izračunu podobnih uporabnikov. V primeru, ko ima uporabnik UX deset zapisov, je 40% od tega štiri kar pomeni, da mora imeti drugi uporabnik Un vsaj štiri enake vsebine.

Verodostojnost je podatek, ki nam govori, koliko zaupamo oceni, ki so jo podali podobni uporabniki za enak izdelek. V primeru, ko so štirje uporabniki in sta oceno za izdelek podala le dva uporabnika, je verodostojnost ocene enaka 50 %. V našem primeru smo postavili parameter verodostojnosti na 60 %.

Skripta začne z delovanjem tako, da najprej poišče uporabnike, ki imajo podporo večjo od 40% – ta podatek preberemo iz nastavitvene tabele v podatkovni bazi. Kot vidimo v tabeli (Tabela 5.3), imamo devet uporabnikov, ki zadostujejo pogoju podpore (U8 ne zadostuje pogoju). Za take uporabnike izračunamo mero podobnosti po standardni Pearsonovi korelaciji (Enačba 5.1). Pearsonov količnik nam pove koliko sta dva uporabnika podobna v razponu od -1 do 1. 1 pomeni, da se popolnoma ujemata in -1 da sta uporabnika čisto nasprotje. V našem primeru so podobni uporabniki vsi, ki imajo Pearsonov količnik večji od 0.5. Aplikacija je v začetnem stanju delovanja in nima dovolj zapisov, da bi si lahko privoščili večji Pearsonov koeficient.

$$r(X, n) = \frac{\sum k(Kx, k - Upx)(Kn, k - Upn)}{\sqrt{\sum k(Kx, k - Ux)^2} * \sqrt{\sum k(Kn, k - Un)^2}} \quad (5.1)$$

Skripta nadaljuje z iskanjem vsebine, ki ima verodostojno oceno večjo od 60% le za uporabnike, ki imajo Pearsonov koeficient večji od 0,5 in podporo večjo od 40 %. Torej v tem primeru, ko imamo devet podobnih uporabnikov s podporo večjo od 40% in samo 4 uporabnika, ki imajo zadosten Pearsonov koeficient. V primeru vidimo, da izdelek K14 nima dovolj ocen podobnih uporabnikov in je njegova verodostojnost le 50%, kar ni dovolj, da bi ga uporabili v naše priporočilne namene. Ko skripta najde podobne izdelke z verodostojno oceno večjo od 60%, nadaljuje z iskanjem ocene posameznega izdelka v razmerju ocene uporabnika, z razliko povprečne ocene posameznega uporabnika.

$$ocena(n) = Up_x + \frac{\sum_{n \in N} (r(X, n)(K_n - Up_n))}{\sum_{n \in N} r(X, n)} \quad (5.2)$$

Enačba nam govori, kako izračunati oceno posameznega izdelka (Enačba 5.2). Kot je razvidno iz formule, seštejemo za en izdelek vse razlike ocen uporabnikov od njegovih povprečnih ocen in na koncu še prištejemo Up_x . Končna ocena, ki jo dobimo, je pravilna ocena izdelka, ki se v nadaljevanju uporabi za izbiro vsebine v priporočilne namene. Kot vidimo iz našega primera izdelek K11 ima najboljše povprečje in ga najprej priporočimo uporabniku.

V primeru, ko procedura iskanja podobnih uporabnikov ne najde dovolj velikega števila vsebine, ki bi jo lahko priporočila uporabniku, se skripta naprej nadaljuje v proceduro iskanja najboljše ocenjenih vsebin posameznih vsebin in nam tam določi potrebne vsebine, ki jih lahko doda na že obstoječi seznam priporočil.

5.1.5 Iskanje najbolj ocenjenih vsebin

Iskanje najbolj ocenjenih vsebin je procedura, ki izračuna oceno za posamezen izdelek. Več ocen za isti izdelek daje boljšo kvaliteto priporočila. Potreba po iskanju povprečja posameznih izdelkov se pojavi, ko uporabnik nima zgodovine. Ker uporabnik nima zgodovine oziroma ima premalo zgodovine, da bi končali z iskanjem vsebine priporočanja, moramo poiskati še dodatne vsebine s proceduro iskanja povprečne vsebine vseh izdelkov.

Vsi izdelki nimajo enakega števila izposoj, zato nam povprečna ocena izdelkov ne pove veliko o tem, kakšen je izdelek. Izdelki, ki so bili večkrat izposojeni, so verjetno bolj priljubljeni. Ne moremo samo seštevati ocen, saj ne bi dobili pravilne skalne lestvice. Na primer, petnajst slabih ocen (1) ni enakovredno kot tri odlične ocene (5). Za takšne in vse ostale primere je potrebno upoštevati uporabnikovo povprečno oceno v vsaki oceni izdelka. Seštevek vseh ocen nam dá končno oceno izdelka.

	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	Upov.
U1	5	5	4	5	5	1	4	3	3	4	5	5	4,08
U2	4	5	3	4	5	1	4	2	3	4	5	4	3,67
U3	3	5	4	2	5	1	4	3	3	4	4	5	3,58
U4	5		5	4	5	1	4	2	3	4	4	5	3,82
U5	4		4	5	5	1	4	2	3	4	5	5	3,82
U6	5		5	5	5	1	4	3	3	4		4	3,90
U7	3		5	4	5	1	4	5	3	4		5	3,90
U8	4		4	3	5	1	4	3	3	4		5	3,60
U9	4		5			1	4		3	4		5	3,71
U10	5					1	4		3	4		4	3,50
U11	4					1	4		3	4		5	3,50
U12	4								3			5	4,00
Povp	4,17	5,00	4,33	4,00	5,00	1,00	4,00	2,88	3,00	4,00	4,60	4,75	
Ocena	4,92	3,67	4,92	1,63	9,63	-30,08	2,92	-7,37	-9,08	2,92	4,03	11,92	

Tabela 5.4: Primer izračuna najboljših ocenjenih izdelkov

Preden se poglobimo v potek delovanja skripte, najprej razložimo osnovne pojme, ki se nahajajo v našem primeru (Tabela 5.4). Kn pomeni knjiga s številom n, Un je uporabnik s številom n, Up nam govori, kakšna je uporabnikova povprečna ocena, Povp. nam prikazuje, kakšna je navadna povprečna ocena izdelka. Slednjega ne uporabljamo v tem izračunu in služi le za prikaz razlik povprečne in dejanske pravilne ocene. Ocena je seštevek vseh razlik ocene in povprečne razlike uporabnika, ki je podal tako oceno.

$$Ocena = \sum_{n \in K} (Kn - Up) \quad (5.3)$$

Zgornja enačba (Enačba 5.3) nam govori, kako pridemo do končne ocene izdelka. Iz formule vidimo, da za več ocen dobimo boljše povprečje oziroma za več slabših ocen potrdimo, da je izdelek res slab. Takšen primer bi bila knjiga K6.

Proces iskanja povprečja posamezne vsebine se prične tako, da skripta poda skozi formulo vsak izdelek posebej in dobi končne ocene. Skala ocen izdelkov je zelo velika in nam prikazuje, kakšna je razlika od najslabše ocenjenega do najbolje ocenjenega izdelka. Uporabniki, ki so ocenili le en tak izdelek, se ne bodo upoštevali v izračunu, saj če je uporabnik podal za ta izdelek oceno 4, ima povprečno oceno 4, ker ni ocenil nobenega drugega izdelka. Razlika je 0, kar pomeni, da njegova ocena ne vpliva na izračun končne ocene izdelka. Vse negativne ocene izdelkov so slabe ocene in takih ne želimo priporočati uporabniku. Cilj priporočilne skripte je uporabniku ponuditi dobre izdelke, da bo še kdaj obiskal knjižnico.

Primer, ko proces iskanja povprečja posamezne vsebine ne najde dovolj vsebin, ki bi jih lahko priporočal uporabniku, se lahko zgodi v začetnem stanju delovanja aplikacije, ko še nimamo dovolj zapisov zgodovine izposojanja. V takem primeru skripta privzame navadne povprečne ocene in z njimi po prioritetenem sistemu sestavi, kaj priporočati uporabnikom.

Zelo pomembno je povedati, da procedura iskanja povprečja posamezne vsebine ocene gradi na nivoju vsebine. V zgornjem primeru smo zaradi boljšega vpogleda na delovanje skripte prikazali izbiro ocene na posamezen izdelek. Skripta torej za vse enake knjige sešteje vse izdelke in poda končno oceno kot oceno vsebine. Za periodiko sešteje vse ocene periodike kot vsebine in v takem primeru priporoča le najnovejši izdelek. Če je tip priporočila medij, ga skripta ocenjuje enako kot knjigo. Sešteje vse ocene vseh medijev znotraj vsebine in priporoča končno oceno vsebine medija.

Po zaključku takšne skripte se zapisi, ki smo jih izbrali za priporočilo uporabniku, pošljejo naprej v tabelo KN_P_PRIPOROCILO, ki se potem uporabi za prikaz priporočenih vsebin na spletni strani. Po takem vstavljanju zapisov v tabelo skripta zaključi z delom. Skripta ponovno sproži delovanje in prične od začetka ob uporabnikovi iteraciji na spletni strani.

Poglavje 6 Sklepne ugotovitve

V diplomskem delu smo najprej pogledali, kako deluje spletna aplikacija »knjižnica«, ki žal še ni na voljo spletnim uporabnikom, dostopna je le članom znotraj društva. Ob pregledu knjižnice smo prišli do novih idej, kako izboljšati delovanje same knjižnice.

Prva naloga izboljšave je bila omogočanje dostopa do aplikacije za širšo javnost. Vpogled v določene vsebine imajo zdaj tako člani kot nečlani v širši javnosti. Ker je aplikacija postala spletno dostopna vsaki osebi, je prav tako prišlo do potrebe po priporočanju vsebine uporabniku. Danes skorajda vsaka večja spletna stran ob obisku uporabnika zgradi določen seznam vsebine, ki naj bi ga zanimal. Zato smo se tudi sami odločili za odločitveni modul, ki bi našim obiskovalcem priporočal najboljše vsebine.

Vsak obiskovalec lahko sam pregleduje vsebino knjižnice in podrobno izve, kakšno je stanje takega zapisa. Da bi zagotovili uporabniku nemoteno uporabo brskanja, smo se odločili dograditi odločitveni modul na posebno, ločeno spletno stran. Obiskovalec ali prijavljeni član, ki ga zanima priporočilo spletne strani, se tako sam prestavi na takšno spletno stran in si ogleda, kar ga zanima.

Za doseganje naših ciljev je bilo potrebno v Oracle podatkovno bazo dograditi dodatne tabele, poglede in podatkovni paket. Ker knjižnica že deluje znotraj Oracle Application Express okolja, smo morali tudi sami uporabiti isto tehnologijo za doseg cilja. Z uporabo PL/SQL programskega jezika je bil zgrajen podatkovni paket, ki vsebuje vse funkcije in procedure za potrebno delovanje priporočilnega modula. Prednost takega paketa je enostavna uporaba in enostavno spreminjanje ob morebitnih nastalih spremembah. Da bo v prihodnosti optimizacija delovanja skripte lažja, smo v podatkovno bazo dodali tudi tabelo z različnimi nastavitvami, ki jih lahko spreminja le skrbnik podatkovne baze.

Ob izdelavi naloge smo prišli tudi do problema različnih vsebin in določili uteži za vsako vsebino posebej. Razlike med vsebinami smo upoštevali tudi v priporočanju, na primer za priporočitev periodike bi bilo smiselno priporočiti le zadnjo izdajo. Da bi bilo delovanje čim boljše, smo morali za vsak izposojen izdelek dodati oceno, ki se potem uporablja v odločitveni skripti.

V nadaljevanju je prišlo tudi do problematike, kako rešiti manjkajoče zapise zgodovine. V aplikaciji imamo različne tipe uporabnikov ter posebej še nove obiskovalce, o katerih, na splošno, ne vemo ničesar. Da bi izboljšali naše priporočilo, smo zgradili več različnih procesov, ki se sprožijo v odvisnosti od tipa uporabnika. S tem skrajšamo delovanje celotne odločitvene skripte.

V sami izdelavi diplomskega dela smo se srečali tudi z izbiro podobnih uporabnikov. Takšna procedura je bila velikega pomena za izdelavo čim bolj natančnega priporočila. V sami izdelavi take procedure smo prišli do sklepov, da ne moremo kar tako seštevati ocen in tako izračunati povprečno oceno. Za takšen izračun ne upoštevamo, kolikokrat so podane ocena in razlike med ocenjevanjem posameznih uporabnikov. Iz tega razloga smo pri vsaki oceni odvzeli povprečno oceno uporabnika in sešteli vse ocene. Takšni povprečni rezultati nam dajo pravo oceno posameznega izdelka.

Pri gradnji spletne strani za prikaz priporočila, smo naleteli na problem hitrosti. Ko je obiskovalec osvežil stran s priporočili, je skripta vsakič na novo računala in ustvarjala priporočila. V primeru, ko je aplikacijo uporabljalo več uporabnikov istočasno, je prišlo do velike upočasnitve delovanja same podatkovne baze kot tudi osvežitve spletne strani. S tem namenom smo v podatkovno bazo dogradili podatkovno tabelo, v kateri se nahajajo zapisi, ki jih priporočamo obiskovalcu spletne strani. Z izdelavo dodatne tabele smo zmanjšali klice procedur po nepotrebnih posegih v podatkovno bazo in tako pospešili njeno delovanje.

Kot v vsakem večjem sistemu se tudi v tem vedno najde del programa oziroma spletne strani, ki bi ga bilo mogoče izboljšati. V izdelavi same diplomske naloge smo prišli do sklepa, da bi bilo zaradi lepšega izgleda potrebno na spletni strani naknadno dodati slike izdelkov. Prav tako je potrebno zgraditi piškotke, da izvemo, če je isti obiskovalec dvakrat ali večkrat obiskal našo spletno stran. Odvečne zapise bi se lahko izbrisalo iz podatkovne baze in s tem tudi pospešilo iskanje podobnih uporabnikov. V odločitveni skripti pa nismo upoštevali časa, ki ga je uporabnik namenil ogledu posamezne vsebine. S tem bi mogoče lahko izboljšali prioritetni sistem izbire priporočila.

Za zaključek je potrebno povedati še, da odločitveni modul trenutno deluje po nastavitvah za začetno delovanje sistema. Po določenem času delovanja aplikacije bo nujno potrebno narediti analizo vseh zapisov in popraviti uteži v podatkovni bazi. S tem se bo izboljšalo tudi samo delovanje odločitvene skripte.

Literatura

- [1] Cyran, M. (2005). *Oracle Database: concepts*. RedwoodCity: Oracle. Pridobljeno 25.10.2014 s spletne strani: http://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
- [2] IZUM. (1997-2014). *Lokalna baza podatkov: Slovenski dom - Kulturno prosvetno društvo 'Bazovica', Rijeka*. Maribor: IZUM. Pridobljeno 25.10.2014 s spletne strani: <http://cobiss4.izum.si/scripts/cobiss?ukaz=BASE&bno=55012&id=1011080767290664>
- [3] IZUM. (2014). *Platforma COBISS – Kooperativni online bibliografski sistem in servisi*. Maribor: IZUM. Pridobljeno 25.10.2014 s spletne strani: http://www.cobiss.net/platforma_cobiss.htm
- [4] Jannach, D. (2010). *Recommendersystems : anintroduction*. Cambridge: Cambridge UniversityPress.
- [5] LeonidasFegaras (1998). *AnotherPLSQL Manual*. Pridobljeno 25.10.2014 s spletne strani: <https://lambda.uta.edu/cse5331/spring98/plsql.html>
- [6] Oracle.com. *Apex*. RedwoodCity: Oracle. Pridobljeno 25.10.2014 s spletne strani: <http://www.oracle.com/technetwork/developer-tools/apex/overview/index-155186.html>
- [7] Oracle.com. *Apex*. RedwoodCity: Oracle. Pridobljeno 25.10.2014 s spletne strani: <http://www.oracle.com/technetwork/testcontent/what-is-apex-099128.html>
- [8] Oracle.com. *Oracle XE*. RedwoodCity: Oracle. Pridobljeno 25.10.2014 s spletne strani: - <http://www.oracle.com/technetwork/database/database-technologies/express-edition/overview/index.html>